SAMPLEWISE RETRAINING COMPATIBILITY: OUTCOME-LEVEL EVALUATION AND GUIDANCE FOR MACHINE UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Approximate machine unlearning is commonly evaluated by fidelity to exact retraining. Existing evaluations either compare a realized unlearned model against a single retraining reference or compare repeated unlearning and retraining outcomes distributionally. The former reduces stochastic retraining to a single reference realization, while the latter characterizes repeated outcomes rather than the single model produced for an individual deletion request. This leaves an outcome-level question: how should a realized unlearned model be evaluated against retraining's natural variability? We formalize *samplewise retraining compatibility (SRC)* to evaluate whether each deleted sample's realized behavior falls within the range naturally produced by stochastic retraining. To operationalize SRC with finite retraining references, we introduce BLORO, which calibrates unlearning deviation against retraining's own variability. To pursue SRC without retraining references, we propose ReCo, which estimates sample-specific proxy targets from difficulty-conditioned nonmember behavior and adaptively controls forgetting toward them. Experiments show that BLORO remains stable with finite retraining references, ReCo achieves strong SRC while remaining competitive on conventional set-level unlearning metrics, and models obtained with ReCo better preserve their post-unlearning behavior on deleted samples during subsequent training on retained data. Together, these results support SRC as a principle for evaluating and guiding approximate unlearning.

1 INTRODUCTION

The General Data Protection Regulation (GDPR) establishes a right to erasure, often referred to as the "right to be forgotten" (Protection Regulation, 2018). When data have already been used for model training, however, deleting the source records does not remove their learned influence. Retraining from scratch on the retained data provides the clearest counterfactual for what a model would have learned without the deleted samples, but is often computationally expensive (Ginart et al., 2019; Bourtoule et al., 2021). Machine unlearning seeks to remove the influence of specified training samples without incurring this full retraining cost, with approximate methods modifying an existing model more efficiently (Guo et al., 2020; Kurmanji et al., 2023; Foster et al., 2024). Their performance is therefore commonly evaluated by fidelity to exact retraining.

Retraining-based evaluations typically take two forms. **Point-reference evaluation** compares an unlearned model with a single retrained model, but exact retraining is stochastic, making the conclusion potentially sensitive to the particular retraining outcome chosen as the reference (Cadet et al., 2025; Lanyon et al., 2025). **Distributional evaluation** instead compares repeated unlearning and retraining outcomes, explicitly accounting for this stochasticity by characterizing the unlearning procedure across repeated executions (Triantafillou et al., 2024). However, this procedure-level view differs from the per-request usage pattern in which an unlearning algorithm ultimately produces one realized model for each deletion request. Existing evaluation therefore leaves a complementary outcome-level question:

> *Given a single realized unlearned model for a deletion request, how should its behavior be evaluated against the natural variability of exact retraining?*

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.

We address this question through Samplewise Retraining Compatibility (SRC). Exact retraining induces a distribution of behavior for each deleted sample, and SRC asks whether that sample's behavior under a single realized unlearned model falls within a plausible region induced by stochastic retraining. We formulate compatibility samplewise because set-level summaries can obscure heterogeneous behavior within the forget set. SRC therefore turns the realized-outcome question into a samplewise membership criterion. Figure 1 summarizes the three evaluation granularities.

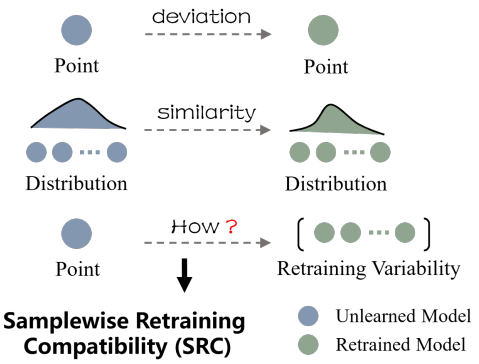

Figure 1: Conceptual comparison of evaluation granularities under stochastic retraining.

Operationalizing this principle raises two practical challenges. First, retraining behavior is observable only through a finite number of retraining realizations. We introduce Balanced Leave One Retraining Out (BLORO), which evaluates an unlearned outcome and held-out retraining outcomes against matched finite reference sets, calibrating unlearning deviation by the deviation that exact retraining itself exhibits under the same reference uncertainty. Second, approximate unlearning cannot rely on exact retraining references to provide sample-specific targets. We therefore develop Retraining-free Control (ReCo), which estimates difficulty-conditioned proxy targets for individual deleted samples from held-out nonmember behavior and adapts forgetting according to each sample's current target violation. The individualized targets determine which samples continue to receive forgetting pressure and jointly provide an automatic stopping condition.

Together, BLORO enables finite-reference outcome-level evaluation without treating any single retrained model as ground truth, while ReCo uses the same compatibility principle to guide unlearning through sample-specific proxy targets without requiring exact retraining references. We further examine whether the resulting samplewise compatibility persists during subsequent training on retained data after the controller is removed. Our contributions are:

- We introduce SRC, an outcome-level criterion for samplewise compatibility with stochastic retraining.
- We propose BLORO, a finite-reference evaluation calibrated by retrain-to-retrain variability.
- We develop ReCo, a retraining-free controller with sample-specific proxy targets and adaptive forgetting.
- Across centralized and federated settings, we show that BLORO remains well calibrated with few retraining references, while ReCo improves samplewise compatibility, remains competitive on conventional set-level metrics, and preserves compatibility under subsequent retained-data training.

2 Related Work

2.1 Machine Unlearning

Machine unlearning aims to remove the influence of specified training samples from a trained model. Exact and certified approaches pursue retraining-equivalent or formally bounded deletion through data partitioning, algorithmic state updates, or certified post-processing (Cao & Yang, 2015; Ginart et al., 2019; Guo et al., 2020; Bourtoule et al., 2021; Neel et al., 2021; Sekhari et al., 2021; Koloskova et al., 2025). Approximate unlearning trades exact guarantees for efficiency through optimization, fine-tuning, distillation, saliency-based updates, or parameter modification (Golatkar et al., 2020; Kurmanji et al., 2023; Foster et al., 2024; Fan et al., 2024; Di et al., 2026). Federated unlearning extends this setting to distributed training under additional communication and data-access constraints (Romandini et al., 2025; Pan et al., 2025). Our work studies approximate unlearning in both centralized and federated settings through the lens of stochastic retraining.

2.2 Evaluating Machine Unlearning

Empirical evaluation commonly combines forget-set behavior, retain/test utility, privacy attacks, and comparisons with exact retraining (Kurmanji et al., 2023; Wichert & Sikdar, 2024; Cadet et al., 2025). Point-reference evaluation compares against a particular retraining outcome and can therefore be sensitive to training randomness (Cadet et al., 2025; Lanyon et al., 2025). Distributional approaches instead compare repeated unlearning and retraining outcomes to characterize algorithm-level fidelity under stochasticity (Triantafillou et al., 2024), while retraining-free evaluation has been studied through alternative statistical criteria (Zhang et al., 2026). Prior distributional evaluations can also assess discrepancies at the individual-sample level, but they characterize repeated unlearning outcomes. SRC instead evaluates the samplewise behavior of a single realized unlearned model against stochastic retraining.

2.3 Sample-Aware Unlearning

Training samples can differ substantially in memorization and unlearning difficulty, motivating sample-aware objectives, reweighting strategies, and adaptive updates (Bărbulescu & Triantafillou, 2024; Zhao et al., 2024; Sepahvand et al., 2025; Cha et al., 2024). These methods demonstrate that uniform forgetting pressure is often insufficient for heterogeneous deletion sets. SRC formalizes the desired post-unlearning behavior of each deleted sample relative to stochastic retraining, while ReCo estimates corresponding sample-specific proxy targets from difficulty-conditioned nonmembers for retraining-free control. FaLW (Yu et al., 2026) already uses unseen-sample distributions for instance-level dynamic loss reweighting. ReCo instead constructs fixed, difficulty-conditioned proxy targets and combines them with samplewise gating and automatic stopping.

3 Samplewise Retraining Compatibility

3.1 Problem Setup

Let $\mathcal{D}$ denote the original training set, $\mathcal{D}_f \subset \mathcal{D}$ the samples requested for deletion, and $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$ the retain set. Let $\mathcal{A}$ denote the randomized training algorithm. The original model is drawn as $\theta^o \sim \mathcal{A}(\mathcal{D})$, while exact retraining after deletion produces $\Theta^R \sim \mathcal{A}(\mathcal{D}_r)$. Importantly, Θ^R is a random variable rather than a fixed checkpoint.

An approximate unlearning method $\mathcal{U}$ instead produces a realized model $\theta^U = \mathcal{U}(\theta^o, \mathcal{D}_f, \mathcal{D}_r)$. Rather than requiring θ^U to match one particular retraining realization, we evaluate its behavior relative to the variability naturally induced by exact retraining.

For a deleted sample $(x_i, y_i) \in \mathcal{D}_f$, we characterize model behavior using the true-label log-odds

$$s_i(\theta) = \log \frac{p_\theta(y_i \mid x_i)}{1 - p_\theta(y_i \mid x_i)}. \tag{1}$$

Larger values indicate stronger support for the true label. We use log-odds because it is monotone in true-label probability, expands the high-confidence region beyond the bounded probability scale, and remains smooth and differentiable for both evaluation and control. SRC is not tied to this particular statistic; alternatives are discussed in the appendix.

3.2 Stochastic Retraining Behavior

For the same deleted sample, valid exact-retraining runs can produce different behavior. Applying the behavioral statistic to the random retrained model gives $S_i^R = s_i(\Theta^R)$, with distribution P_i^R over retraining randomness. Both the location and variability of P_i^R can differ across deleted samples, motivating a sample-specific rather than global notion of compatible forgetting.

Definition 1 (Samplewise Retraining Compatibility). Let $\mathcal{C}_i^R$ denote a plausible behavioral region induced by P_i^R, representing the range of outcomes considered plausible under stochastic exact retraining. A realized unlearned model θ^U is retraining compatible for sample i if

$$s_i(\theta^U) \in \mathcal{C}_i^R. \tag{2}$$

For a scalar statistic oriented so that larger values indicate stronger true-label support, write $\mathcal{C}_i^R = [L_i^R, U_i^R]$. The realized outcome is then classified as

$$\mathrm{State}_i(\theta^U) = \begin{cases} \text{over-forgotten}, & s_i(\theta^U) < L_i^R, \\ \text{compatible}, & L_i^R \le s_i(\theta^U) \le U_i^R, \\ \text{under-forgotten}, & s_i(\theta^U) > U_i^R. \end{cases} \tag{3}$$

SRC is therefore a samplewise criterion on a single realized unlearning outcome; it does not require the distribution of unlearning outcomes to equal the retraining distribution. The two incompatible states distinguish excessive from insufficient forgetting relative to each sample's own retraining behavior.

Definition 1 is an idealized population criterion because P_i^R is not directly observable. Figure 2 provides an overview of SRC and its two operationalizations: BLORO for finite-reference evaluation and ReCo for retraining-free compatibility guidance.

4 BLORO: Retraining-Calibrated Evaluation

The stochastic retraining distribution P_i^R is not directly observable, so SRC must be estimated from finitely many retraining realizations. With few references, even a genuine retraining outcome may fall outside the empirical range of the others. BLORO therefore calibrates unlearning deviations against the deviations exhibited by held-out retraining outcomes under matched finite-reference uncertainty, as illustrated in the left panel of Fig. 2.

4.1 Balanced Leave-One-Retraining-Out

Suppose a deletion request has K independent retraining references $R^{(1)}, \ldots, R^{(K)} \sim \mathcal{A}(\mathcal{D}_r)$. For fold k, BLORO holds out $R^{(k)}$ as a retrain-as-query outcome and uses

$$\mathcal{R}_{-k} = \{R^{(j)} : j \neq k\}$$

as the common reference set. For sample i, let $\mu_{i,-k}$ and $\sigma_{i,-k}$ denote the mean and standard deviation of $\{s_i(R^{(j)}) : j \neq k\}$. For any query model M, define

$$z_{ik}(M) = \frac{s_i(M) - \mu_{i,-k}}{\sigma_{i,-k} + \epsilon}, \tag{4}$$

where ϵ is a numerical stabilizer.

We then compute

$$z_{ik}^U = z_{ik}(\theta^U), \qquad z_{ik}^R = z_{ik}(R^{(k)}).$$

The comparison is balanced because the unlearned and held-out retraining queries are evaluated against the same $K-1$ references. Thus, z_{ik}^R captures the deviation that genuine retraining itself exhibits under finite references, while z_{ik}^U measures the unlearned outcome on the same scale.

4.2 Retrain-as-Query Calibration

Let $\mathcal{Z}_R^{\mathrm{cal}}$ collect z_{ik}^R values from other deletion requests used for calibration, excluding the request being evaluated. Given a target retraining coverage $1-\alpha$, define

$$\tau_L = Q_{\alpha/2}\big(\mathcal{Z}_R^{\mathrm{cal}}\big), \qquad \tau_U = Q_{1-\alpha/2}\big(\mathcal{Z}_R^{\mathrm{cal}}\big). \tag{5}$$

An unlearned outcome is compatible for sample i under fold k when

$$\tau_L \le z_{ik}^U \le \tau_U,$$

with values above τ_U indicating under-forgetting and values below τ_L indicating over-forgetting. Since z_{ik} is standardized using sample i's own retraining references, these shared thresholds induce the sample- and fold-specific region

$$\widehat{\mathcal{C}}_{ik}^R = [\mu_{i,-k} + (\sigma_{i,-k} + \epsilon)\tau_L,\ \mu_{i,-k} + (\sigma_{i,-k} + \epsilon)\tau_U], \tag{6}$$

which is the finite-reference counterpart of $\mathcal{C}_i^R$ in Definition 1. Calibration excludes the evaluated request; further details are provided in Appendix B.

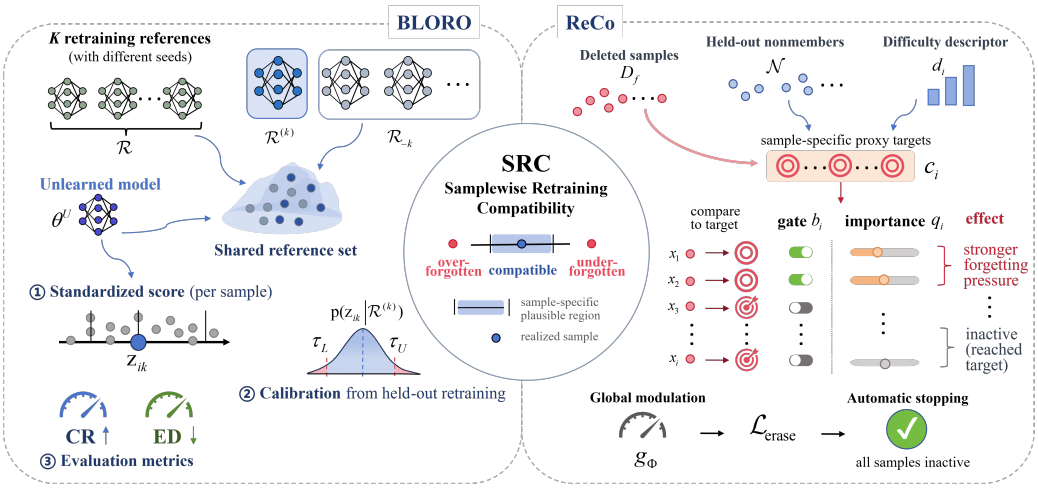

Figure 2: Overview of the proposed framework. Center: Samplewise Retraining Compatibility (SRC). Left: BLORO for retraining-calibrated evaluation. Right: ReCo for retraining-free compatibility control.

4.3 Compatibility Metrics

We summarize samplewise compatibility using two metrics. The *Compatibility Rate (CR)* measures the fraction of compatible sample–fold pairs:

$$\mathrm{CR} = \frac{1}{|\mathcal{D}_f|K} \sum_{i=1}^{|\mathcal{D}_f|} \sum_{k=1}^{K} \mathbf{1}\left[\tau_L \le z_{ik}^U \le \tau_U\right]. \tag{7}$$

The *Excess Deviation (ED)* measures the magnitude of violations:

$$\mathrm{ED} = \frac{1}{|\mathcal{D}_f|K} \sum_{i=1}^{|\mathcal{D}_f|} \sum_{k=1}^{K} \left([z_{ik}^U - \tau_U]_+ + [\tau_L - z_{ik}^U]_+\right), \tag{8}$$

where $[a]_+ = \max(a, 0)$. Higher CR and lower ED indicate stronger samplewise retraining compatibility. Because the thresholds are calibrated from genuine retrain-as-query deviations, exact retraining is expected to attain approximately the target coverage $1 - \alpha$, rather than $\mathrm{CR} = 1$. Directional under- and over-forgetting rates are defined in Appendix B.2.

5 ReCo: Compatibility Control without Retraining

BLORO uses exact retraining references for offline evaluation, whereas such references are unavailable during approximate unlearning. As illustrated in the right panel of Fig. 2, we therefore introduce ReCo, which estimates sample-specific proxy targets from difficulty-conditioned nonmembers and uses them to adaptively control which deleted samples continue receiving forgetting pressure and when unlearning should stop. In federated settings, the procedure is performed locally at the forgetting client using its deletion samples and labeled held-out nonmembers; retraining-free control still requires this auxiliary data.

5.1 Samplewise Proxy Targets

Let $\mathcal{N}$ be a held-out nonmember set never used to train the original model. For each deleted sample and nonmember, we compute the behavioral score s from Eq. (1) and a robust original-model difficulty descriptor d derived from deterministic views (Appendix C.1).

Using only $\mathcal{N}$, ReCo estimates the conditional median $\widehat{Q}_{0.5}(s \mid d)$ and assigns each deleted sample the fixed target

$$c_i = \widehat{Q}_{0.5}(s \mid d_i) + \Delta, \tag{9}$$

where Δ is a global residual correction estimated from a disjoint nonmember split. This construction assumes that, after conditioning on original-model difficulty, nonmember behavior approximates the central retraining behavior of deleted samples. We formalize this conditional-transfer assumption and empirically evaluate the resulting proxy-target accuracy in Appendix C.2. All targets are computed before unlearning and remain fixed thereafter.

5.2 Target-Based Gating and Adaptive Control

At step t, ReCo measures each sample's normalized target violation and activity as

$$m_i(\theta_t) = \frac{s_i(\theta_t) - c_i}{a}, \qquad b_i = \mathbf{1}[m_i(\theta_t) > 0], \tag{10}$$

where $a > 0$ is a robust scale estimated from the nonmember calibration set. Once $s_i(\theta_t) \le c_i$, the sample becomes inactive and receives no further direct forgetting pressure.

Among active samples, ReCo prioritizes those contributing more strongly to the current request-level misalignment. Let Φ be a piecewise differentiable potential capturing both typical and large target violations. We define

$$q_i = \frac{\partial \Phi}{\partial s_i}, \qquad g_\Phi = \sigma(\Phi/T), \qquad \widetilde{q}_i = g_\Phi \, b_i \, q_i, \tag{11}$$

where q_i determines the relative priority of sample i and g_Φ scales the overall update according to the request-level violation. The resulting erasure objective is

$$\mathcal{L}_{\mathrm{erase}}(\theta) = \sum_{i \in \mathcal{D}_f} \mathrm{sg}(\widetilde{q}_i) \, s_i(\theta), \tag{12}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. Thus, only samples still above their targets receive direct forgetting pressure, with larger violations receiving greater priority. The construction of Φ and the detached-weight implementation are given in Appendix C.3.

5.3 Automatic Stopping and Stable Updates

After each accepted update, ReCo re-evaluates the complete deletion set and terminates once all samples remain inactive for a fixed number of consecutive checks. Adaptive step scaling, parameter-drift constraints, and backtracking against Φ limit overshoot near the targets; implementation details are provided in Appendix C.4. This is one-sided control toward proxy targets, not enforcement of SRC's two-sided regions: updates driven by other samples can still move an inactive sample below its compatible range.

6 Experiments

We organize the experiments around four questions: whether stochastic retraining and sample heterogeneity motivate SRC, whether BLORO remains well calibrated with few retraining references, whether SRC-guided control improves realized unlearning outcomes, and whether the resulting compatibility persists under subsequent retained-data training.

6.1 Experimental Setup

Datasets, Models, and Settings. We evaluate sample-level unlearning on the CIFAR-10 and CIFAR-100 datasets (Krizhevsky, 2009) in both centralized and federated settings. All experiments use ResNet-18 (He et al., 2016). Each deletion request contains 100 randomly sampled training examples; the federated setting uses five IID clients with full participation, with the requested samples drawn from client 0. Centralized models are trained with SGD and federated models with FedAvg (McMahan et al., 2017). Full training protocols, data splits, and implementation details are provided in Appendix A.

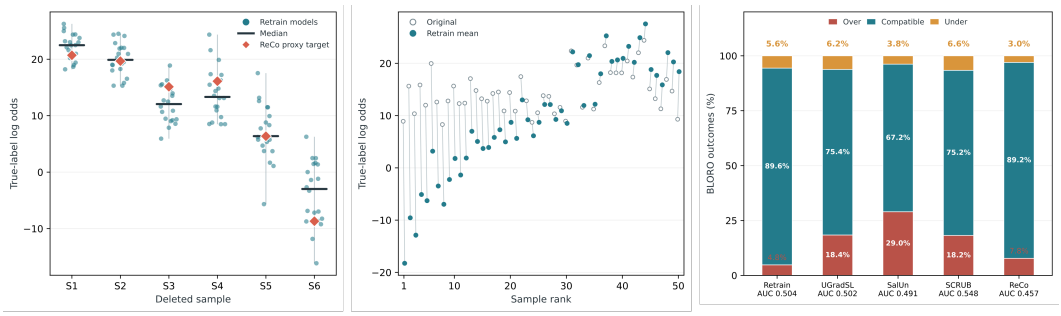

Figure 3: Empirical motivation for SRC on CIFAR-10. (a) Retraining variability and ReCo proxy targets. (b) Sample-specific retraining displacement. (c) Different samplewise outcomes under similar LiRA AUCs.

Baselines and metrics. For centralized unlearning, we compare ReCo with generic retained-data fine-tuning (FT), clipped GA (Graves et al., 2021; Thudi et al., 2022), SCRUB (Kurmanji et al., 2023), SalUn (Fan et al., 2024), SFR-on (Huang et al., 2024), UGradSL (Di et al., 2026), and Sharp MinMax (Tang & Khanna, 2026). For federated unlearning, FedFT and clipped FedGA adapt FT and GA to FedAvg; we additionally compare with FUPGA (Halimi et al., 2022), EWCSGA (Wu et al., 2022), and FedOSD (Pan et al., 2025), evaluated at its unlearning-stage endpoint without the method-specific projected recovery. Exact Retrain and FedRetrain are included as reference outcomes rather than approximate methods. We report forget accuracy and test accuracy for forgetting and utility, online LiRA AUC (Carlini et al., 2022) for membership privacy, and BLORO-based Compatibility Rate (CR, higher is better) and Excess Deviation (ED, lower is better) for samplewise retraining compatibility. Unless otherwise specified, BLORO uses $K = 5$ retraining references and target retraining coverage $1 - \alpha = 0.9$.

Comparison protocol. For each dataset and setting, we use two independently trained original models and three deletion requests per model, giving six request-level evaluations. Methods evaluated on the same request share the original checkpoint, data partition, deletion set, privacy-audit bank, and retraining references. All method configurations, including ReCo and the baselines, are fixed before evaluation on the six formal requests; neither CR nor ED is used for hyperparameter or checkpoint selection. Tables report the mean with sample standard deviation in parentheses across the six requests as descriptive summaries; requests sharing an original model are not fully independent, and no statistical significance claim is made. Additional implementation and tuning details are provided in the appendix.

6.2 Empirical Motivation for Samplewise Retraining Compatibility

We first examine the empirical phenomena motivating SRC using a centralized CIFAR-10 deletion request with 20 independent exact-retraining realizations. The behavioral statistic is the true-label log-odds $s_i(\theta)$ from Eq. (1).

Figure 3 reveals three effects. First, the same deleted sample exhibits non-negligible variation across valid retraining realizations, confirming that exact retraining induces a distribution rather than a unique target; the ReCo proxy targets in Fig. 3(a) also typically lie near the corresponding retraining behavior, supporting their use as sample-specific guidance. Second, both the location and required displacement of retraining behavior vary across samples. Finally, methods with similar set-level LiRA AUCs can exhibit markedly different compositions of compatible, under-forgotten, and over-forgotten samples. Together, these observations motivate evaluating unlearning relative to each sample's own stochastic retraining behavior rather than a single reference or aggregate forget-set statistic.

6.3 Finite-Reference Calibration of BLORO

Table 1: Centralized unlearning results at the unlearning endpoint. Mean (SD) over six deletion requests.

	CIFAR-10					CIFAR-100				
Method	Forget Acc.	Test Acc.	LiRA AUC	CR ↑	ED ↓	Forget Acc.	Test Acc.	LiRA AUC	CR ↑	ED ↓
Retrain	0.845(0.034)	0.831(0.004)	0.492(0.036)	0.899(0.007)	0.172(0.047)	0.737(0.038)	0.712(0.003)	0.527(0.037)	0.898(0.014)	0.160(0.036)
FT	1.000(0.000)	0.820(0.003)	0.701(0.039)	0.712(0.052)	0.619(0.188)	0.788(0.046)	0.637(0.006)	0.634(0.029)	0.686(0.042)	0.956(0.215)
GA (clip)	0.982(0.008)	0.797(0.005)	0.730(0.039)	0.775(0.047)	0.568(0.146)	0.183(0.170)	0.156(0.142)	0.535(0.036)	0.211(0.191)	10.608(6.991)
SCRUB	0.860(0.024)	0.808(0.008)	0.551(0.041)	0.709(0.034)	0.827(0.319)	0.730(0.042)	0.690(0.008)	0.608(0.051)	0.609(0.043)	1.633(0.211)
SalUn	0.850(0.051)	0.805(0.005)	0.450(0.039)	0.619(0.036)	1.057(0.247)	0.750(0.023)	0.703(0.001)	0.489(0.031)	0.548(0.069)	1.735(0.336)
SFR-on	0.392(0.241)	0.789(0.025)	0.391(0.116)	0.309(0.214)	6.198(3.249)	0.688(0.047)	0.672(0.014)	0.578(0.051)	0.674(0.044)	1.076(0.243)
UGradSL	0.808(0.022)	0.750(0.020)	0.529(0.044)	0.654(0.057)	0.914(0.294)	0.477(0.133)	0.526(0.139)	0.531(0.034)	0.493(0.110)	3.338(1.471)
Sharp MinMax	0.592(0.026)	0.785(0.011)	0.430(0.036)	0.456(0.066)	2.006(0.663)	0.620(0.061)	0.668(0.008)	0.503(0.027)	0.650(0.120)	1.166(0.575)
ReCo	0.817(0.066)	0.785(0.012)	0.482(0.028)	0.805(0.058)	0.391(0.132)	0.695(0.049)	0.684(0.006)	0.538(0.038)	0.783(0.031)	0.453(0.084)

Table 2: Federated unlearning results at the unlearning endpoint. Mean (SD) over six deletion requests. EWCSGA's CIFAR-10 ED includes a finite 7.74×10^{17} request-level value.

	CIFAR-10					CIFAR-100				
Method	Forget Acc.	Test Acc.	LiRA AUC	CR ↑	ED ↓	Forget Acc.	Test Acc.	LiRA AUC	CR ↑	ED ↓
FedRetrain	0.822(0.047)	0.813(0.005)	0.493(0.027)	0.900(0.004)	0.175(0.028)	0.654(0.035)	0.651(0.003)	0.496(0.023)	0.899(0.016)	0.168(0.031)
FedFT	1.000(0.000)	0.811(0.002)	0.768(0.039)	0.710(0.043)	0.635(0.179)	0.645(0.170)	0.526(0.087)	0.594(0.058)	0.725(0.063)	0.958(0.630)
FedGA (clip)	0.983(0.005)	0.807(0.003)	0.733(0.042)	0.718(0.047)	0.610(0.178)	0.600(0.048)	0.563(0.054)	0.564(0.027)	0.642(0.040)	1.008(0.288)
FUPGA	0.742(0.210)	0.694(0.104)	0.550(0.116)	0.552(0.133)	2.127(1.293)	0.365(0.191)	0.404(0.209)	0.478(0.023)	0.572(0.115)	2.033(1.302)
EWCSGA	0.533(0.490)	0.454(0.388)	0.587(0.129)	0.358(0.394)	1.29×10^{17} (3.16×10^{17})	0.510(0.232)	0.478(0.216)	0.553(0.043)	0.593(0.232)	2.161(2.928)
FedOSD	0.785(0.053)	0.798(0.007)	0.608(0.033)	0.683(0.061)	0.765(0.316)	0.537(0.063)	0.610(0.037)	0.529(0.040)	0.743(0.026)	0.685(0.119)
ReCo	0.780(0.056)	0.779(0.010)	0.473(0.059)	0.748(0.037)	0.595(0.110)	0.558(0.061)	0.633(0.008)	0.460(0.020)	0.771(0.022)	0.531(0.089)

We vary the reference-set size $K \in \{3, 5, 10, 15\}$ and compare BLORO with a direct empirical interval constructed from the available retraining references. BLORO thresholds are calibrated from another deletion request, using the same out-of-request principle as the main evaluation.

Figure 4 shows that the direct empirical interval under-covers genuine retraining outcomes with few references and approaches the nominal level as K increases. In contrast, BLORO achieves near-nominal coverage even at small K, demonstrating that retrain-as-query calibration enables reliable aggregate SRC evaluation with limited retraining references.

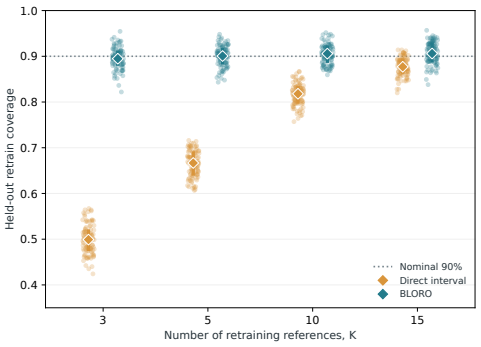

Figure 4: Held-out retraining coverage versus reference-set size K. Dotted line: nominal 90% coverage.

6.4 Main Comparison at the Unlearning Endpoint

Tables 1 and 2 compare conventional set-level metrics with BLORO-based samplewise compatibility in centralized and federated settings. Exact retraining is expected to attain CR near the nominal coverage $1 - \alpha$, rather than one.

Across both settings, ReCo achieves the highest mean CR and lowest reported mean ED among the compared approximate methods, while maintaining LiRA AUC close to exact retraining. Notably, conventional set-level metrics do not fully reflect samplewise compatibility. For example, on centralized CIFAR-10, SalUn achieves a LiRA AUC of 0.450, compared with 0.482 for ReCo, but obtains substantially lower CR (0.619 vs. 0.805) and higher ED (1.057 vs. 0.391). This illustrates how conventional privacy metrics can obscure deviations from samplewise retraining behavior. These compatibility gains come with lower test accuracy at the unlearning endpoint. We next examine whether subsequent retained-data training can recover utility while preserving compatibility.

Table 3: CIFAR-10 results after retained-data training without unlearning control. Mean (SD) over six deletion requests.

Centralized					Federated				
Method	Test Acc.	LiRA AUC	CR ↑	ED ↓	Method	Test Acc.	LiRA AUC	CR ↑	ED ↓
Retrain	0.831(0.004)	0.492(0.036)	0.899(0.007)	0.172(0.047)	FedRetrain	0.813(0.005)	0.493(0.027)	0.900(0.004)	0.175(0.028)
SCRUB	0.810(0.006)	0.556(0.039)	0.721(0.039)	0.802(0.340)	FUPGA	0.761(0.046)	0.559(0.122)	0.575(0.113)	1.744(1.003)
SalUn	0.812(0.005)	0.463(0.040)	0.656(0.045)	0.838(0.272)	FedOSD	0.809(0.001)	0.622(0.036)	0.733(0.047)	0.593(0.279)
ReCo	0.817(0.004)	0.532(0.038)	0.850(0.034)	0.279(0.101)	ReCo	0.807(0.002)	0.511(0.055)	0.810(0.017)	0.399(0.129)

6.5 Ablation of ReCo

Table 4 examines three key components of ReCo. Replacing sample-specific targets with a global target reduces CR from 0.805 to 0.571, highlighting the importance of individualized forgetting targets. Removing target-based gating causes the largest degradation, reducing CR to 0.506 and increasing ED to 2.003. Uniform weighting also weakens compatibility, although its effect is comparatively modest. Notably, similar LiRA AUC values can accompany substantially different compatibility outcomes, further demonstrating that conventional privacy metrics do not fully capture samplewise retraining compatibility.

Table 4: ReCo ablations on centralized CIFAR-10. Mean (SD) over six deletion requests.

Variant	LiRA AUC	CR ↑	ED ↓
Full ReCo	0.482(0.028)	0.805(0.058)	0.391(0.132)
Global target	0.588(0.054)	0.571(0.053)	1.311(0.625)
Uniform weighting	0.475(0.031)	0.781(0.065)	0.463(0.159)
No gating	0.531(0.048)	0.506(0.159)	2.003(1.445)

6.6 Persistence under Subsequent Retained-Data Training

As a targeted persistence study on CIFAR-10, we continue retained-data training from the unlearning endpoints of ReCo and selected baselines in both centralized and federated settings, with no unlearning-specific control active during this stage. All compared endpoints use the same retained-data training protocol; this stage does not reproduce FedOSD's native projected recovery.

Table 3 shows that subsequent retained-data training recovers much of ReCo's endpoint utility while further improving samplewise compatibility in both settings. Its CR increases from 0.805 to 0.850 in the centralized setting and from 0.748 to 0.810 in the federated setting, with corresponding reductions in ED. Although selected baselines also recover utility, their post-training compatibility remains lower than that of ReCo. Importantly, these gains occur without continued controller intervention, indicating that ReCo's compatibility advantage persists under the evaluated retained-data training protocol.

6.7 Discussion

These results support SRC as a common principle for evaluating and guiding approximate unlearning. Conventional set-level metrics can obscure samplewise deviations from stochastic retraining, while SRC explicitly captures such heterogeneity. Moreover, the persistence results suggest that guiding unlearning toward retraining-compatible behavior may yield model states that better preserve compatibility during subsequent retained-data training.

7 Conclusion

We introduce samplewise retraining compatibility (SRC) as an outcome-level principle for evaluating and guiding approximate machine unlearning relative to stochastic exact retraining, instantiated through BLORO for evaluation and ReCo for retraining-free guidance. Experiments across centralized and federated settings show that SRC reveals samplewise deviations hidden by conventional metrics and promotes retraining-aligned behavior that persists under subsequent retained-data training.

AI USE STATEMENT

Generative AI tools were used for language polishing and consistency checks, literature retrieval and discovery, and assistance in implementing experimental frameworks and research code based on author-designed methods. Suggested literature was verified against the original sources, and AI-assisted code was reviewed and tested by the authors. All AI-assisted content was reviewed by the authors, who take full responsibility for the final manuscript and reported results.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we provide the experimental code as part of the supplementary material. We also report detailed experimental protocols, implementation settings, baseline configurations, and evaluation procedures in the appendix.

REFERENCES

George-Octavian Bărbulescu and Peter Triantafillou. To each (Textual sequence) its own: Improving memorized-data unlearning in large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 3003–3023, 2024.

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, 2021.

Xavier F. Cadet, Anastasia Borovykh, Mohammad Malekzadeh, Sara Ahmadi-Abhari, and Hamed Haddadi. Deep unlearn: Benchmarking machine unlearning for image classification. In *2025 IEEE 10th European Symposium on Security and Privacy*, pp. 939–962, 2025.

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 Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy*, pp. 1897–1914, 2022. doi: 10.1109/SP46214.2022.9833649.

Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11186–11194, 2024. doi: 10.1609/aaai.v38i10.28996.

Zonglin Di, Zhaowei Zhu, Jinghan Jia, Jiancheng Liu, Zafar Takhirov, Bo Jiang, Yuanshun Yao, Sijia Liu, and Yang Liu. Label smoothing improves machine unlearning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2d19debb2ea6a4620f7fe67372cca800-Abstract-Conference.html.

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. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/ec4d2e436794d1bf55ca83f5ebb31887-Abstract-Conference.html.

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

Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Y. 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9304–9312, 2020.

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

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3832–3842, 2020.

Anisa Halimi, Swanand Kadhe, Ambrish Rawat, and Nathalie Baracaldo. Federated unlearning: How to efficiently erase a client in FL? In *ICML 2022 Workshop on Updatable Machine Learning*, 2022. URL https://arxiv.org/abs/2207.05521.

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.

Zhehao Huang, Xinwen Cheng, JingHao Zheng, Haoran Wang, Zhengbao He, Tao Li, and Xiaolin Huang. Unified gradient-based machine unlearning with remain geometry enhancement. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0831.

Anastasia Koloskova, Youssef Allouah, Animesh Jha, Rachid Guerraoui, and Sanmi Koyejo. Certified unlearning for neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31275–31298, 2025.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jamie Lanyon, Axel Finke, Petros Andreou, and Georgina Cosma. On the limitation of evaluating machine unlearning using only a single training seed. *arXiv preprint arXiv:2510.26714*, 2025.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, 2017.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pp. 931–962, 2021.

Zibin Pan, Zhichao Wang, Chi Li, Kaiyan Zheng, Boqi Wang, Xiaoying Tang, and Junhua Zhao. Federated unlearning with gradient descent and conflict mitigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19804–19812, 2025. doi: 10.1609/aaai.v39i19.34181.

Protection Regulation. General data protection regulation. *Intouch*, 25:1–5, 2018.

Nicolò Romandini, Alessio Mora, Carlo Mazzocca, Rebecca Montanari, and Paolo Bellavista. Federated unlearning: A survey on methods, design guidelines, and evaluation metrics. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):11697–11717, 2025. doi: 10.1109/TNNLS.2024.3478334.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Nazanin Mohammadi Sepahvand, Anvith Thudi, Berivan Isik, Ashmita Bhattacharyya, Nicolas Papernot, Eleni Triantafillou, Daniel M. Roy, and Gintare Karolina Dziugaite. Leveraging per-instance privacy for machine unlearning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53906–53922, 2025.

Haoran Tang and Rajiv Khanna. Sharpness-aware machine unlearning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/815d50638e46b7f3bdaaae2158c68d54-Abstract-Conference.html.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling SGD: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy*, 2022. doi: 10.1109/EuroSP53844.2022.00027.

Eleni Triantafillou, Peter Kairouz, Fabian Pedregosa, Jamie Hayes, Meghdad Kurmanji, Kairan Zhao, Vincent Dumoulin, Julio Jacques Junior, Ioannis Mitliagkas, Jun Wan, Lisheng Sun Hosoya, Sergio Escalera, Gintare Karolina Dziugaite, Peter Triantafillou, and Isabelle Guyon. Are we making progress in unlearning? findings from the first NeurIPS unlearning competition. *arXiv preprint arXiv:2406.09073*, 2024.

Leon Wichert and Sandipan Sikdar. Rethinking evaluation methods for machine unlearning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4727–4739. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.271. URL https://aclanthology.org/2024.findings-emnlp.271/.

Leijie Wu, Song Guo, Junxiao Wang, Zicong Hong, Jie Zhang, and Yaohong Ding. Federated unlearning: Guarantee the right of clients to forget. *IEEE Network*, 36(5):129–135, 2022. doi: 10.1109/MNET.001.2200198.

Liheng Yu, Zhe Zhao, Yuxuan Wang, Pengkun Wang, Xiaofeng Cao, Binwu Wang, and Yang Wang. FaLW: A forgetting-aware loss reweighting for long-tailed unlearning. In *International Conference on Learning Representations*, 2026.

Chenhao Zhang, Muxing Li, Feng Liu, Weitong Chen, and Miao Xu. Unlearning evaluation through subset statistical independence. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d82edced9b8574121960c443d8977267-Abstract-Conference.html.

Kairan Zhao, Meghdad Kurmanji, George-Octavian Barbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

A Experimental Protocol and Supplementary Results

This appendix provides the data, training, and evaluation protocols needed to reproduce the experiments, followed by the calibration details of BLORO and the implementation of ReCo. Supplementary analyses focus on CIFAR-10; the training configuration of CIFAR-100 is included here for completeness.

A.1 Datasets, Training Worlds, and Deletion Requests

Both CIFAR-10 and CIFAR-100 contain 50,000 training and 10,000 test images. Centralized experiments hold out 10% of the training split for validation, leaving 45,000 examples for original-model training. Federated experiments partition the 50,000 training examples IID across five clients and use full participation in every FedAvg round. Each deletion request contains 100 training examples, sampled from the centralized training split or from client 0 in the federated setting. The retained training sets consequently contain 44,900 examples centrally or 9,900 examples at client 0.

We train two original-model worlds (seeds 2027 and 2028) and sample three deletion requests per world. Request seeds are the corresponding world seed plus 203, 204, and 205. All methods within a request share the original checkpoint, data partitions, forget set, privacy-audit bank, and exact-retraining references. Five primary retraining models are trained for each request using training seeds equal to the world seed plus 2000–2004; the same training recipe is used as for the original model, with requested examples removed and training randomness changed. Thus, each setting has six request-level evaluations.

ReCo uses labeled nonmembers held out from original-model training. In the centralized setting, 960 examples are reserved from the validation split; in the federated setting, 960 examples are reserved from client 0's local test partition to match its class proportions. The reserves are excluded from membership-audit negatives and from the federated test-accuracy evaluation. Each reserve is partitioned into three disjoint 320-example pools (A/B/C), of which 80 examples per pool are active in the main run: 160 for conditional fitting and selection and 80 for residual calibration. Centralized reserve selection uses the request seed plus 271; federated reserve selection uses the world seed plus 271. The active-pool shuffles use the request seed plus 890 centrally and the world seed plus 896 federally. Centralized test accuracy uses the official 10,000 images; federated test accuracy uses the remaining 9,040 images. Retain accuracy is measured on at most 8,000 retained training examples.

A.2 Model Training and Computing Environment

All experiments use a CIFAR-style ResNet-18 with a stride-1 3×3 stem, no max pooling, and GroupNorm with 16 channels per group (4, 8, 16, and 32 groups in the four stages). Each stage has two basic residual blocks, followed by global average pooling and a dataset-specific classifier. Table 5 gives the original and exact-retraining recipes for both datasets.

Table 5: Frozen original-model training configurations. "Local Ep." denotes local epochs per FedAvg round.

Dataset	Setting	Epochs/Rounds	Local Ep.	Batch	LR	Momentum	WD	Schedule
CIFAR-10	Centralized	80	–	128	0.02	0.9	0	fixed
CIFAR-10	Federated	40	2	128	0.02	0.9	0	fixed
CIFAR-100	Centralized	160	–	256	0.10	0.9	5×10^{-4}	cosine
CIFAR-100	Federated	160	1	128	0.05	0.9	5×10^{-4}	cosine

CIFAR-10 uses no stochastic augmentation. Its normalization parameters are $\mu = (0.4914, 0.4822, 0.4465)$ and $\sigma = (0.2470, 0.2435, 0.2616)$. CIFAR-100 uses padding-4 random crops to 32×32 and random horizontal flips, with normalization parameters $\mu = (0.5071, 0.4867, 0.4408)$ and $\sigma = (0.2675, 0.2565, 0.2761)$. Experiments were run on a server with two NVIDIA GeForce RTX 5090 GPUs (32 GB per GPU), Python 3.10.12, and PyTorch 2.9.0+cu128.

Table 6: Frozen CIFAR-10 baseline budgets and distinguishing hyperparameters. All methods use the shared request assets.

Method	Budget	Settings
FT	10 epochs; batch 128	LR 0.01, momentum 0.9, WD 5×10^{-4}
GA (clip)	30 steps; batch 128	LR 1, gradient norm cap 0.01, momentum/WD 0
SCRUB	3 epochs; retain/forget batches 128/32	LR 0.005, $T = 4$, $\gamma = 0.99$, $\alpha = 0.001$
SalUn	5 epochs; batch 256	LR 0.0013, salient fraction 0.5
SFR-on	1,500 iterations; batch 128	LR 0.01 cosine, forget every 5, $\alpha_f = 10^4$
UGradSL	9 epochs; batch 256	LR 0.05, retain ratio 0.9, smoothing -0.4
Sharp MinMax	8 epochs; batch 256	LR 0.02, retain weight 0.99, mask 0.30, $\rho = 1$
FedFT	20 full-participation rounds	512/client/round, batch 128, LR 0.005
FedGA (clip)	10 steps; batch 128	LR 1, gradient norm cap 0.01
FUPGA	5 ascent passes; batch 1,024	LR 0.1, momentum 0.9, norm cap 5
EWCSGA	2 passes; batch 10	LR 0.025, decay 0.999, $\lambda_{\mathrm{EWC}} = 10^6$, no clip
FedOSD (unlearning stage)	100 rounds; batch 200	LR 0.04, decay 0.999, 512 retained/client/round

Table 7: Frozen CIFAR-100 baseline budgets and distinguishing hyperparameters. All methods use the shared request assets.

Method	Budget	Settings
FT	10 epochs; batch 128	LR 0.01, momentum 0.9, WD 5×10^{-4}
GA (clip)	22 steps; batch 128	LR 1, gradient norm cap 0.03, momentum/WD 0
SCRUB	5 epochs; retain/forget batches 128/32	LR 0.005, $T = 4$, $\gamma = 0.99$, $\alpha = 0.001$
SalUn	8 epochs; batch 256	LR 0.0013, salient fraction 0.5
SFR-on	150 iterations; batch 128	LR 0.01 cosine (horizon 750), forget every 5, $\alpha_f = 10^4$
UGradSL	13 epochs; batch 256	LR 0.05, retain ratio 0.9, smoothing -0.4
Sharp MinMax	17 epochs; batch 256	LR 0.006, retain weight 0.99, mask 0.30, $\rho = 1$
FedFT	18 full-participation rounds	512/client/round, batch 128, LR 0.05
FedGA (clip)	22 steps; batch 128	LR 1, gradient norm cap 0.02
FUPGA	5 ascent passes; batch 1,024	LR 0.05, momentum 0.9, norm cap 5
EWCSGA	5 passes; batch 10	LR 0.005, decay 0.999, $\lambda_{\mathrm{EWC}} = 10^3$, ascent norm cap 5
FedOSD (unlearning stage)	6 rounds; batch 200	LR 0.04, decay 0.999, 512 retained/client/round

A.3 Baseline Protocols and Hyperparameters

The centralized baselines are FT, clipped GA, SCRUB, SalUn, SFR-on, UGradSL, and Sharp MinMax; the federated baselines are FedFT, clipped FedGA, FUPGA, EWCSGA, and FedOSD. Exact retraining is included as a reference, not as an approximate-unlearning baseline. We fixed all method configurations before evaluating the six formal requests. Adaptations to random-100 deletion were selected on a separate development request, primarily using forget accuracy and class-matched proxy MIA AUC, with retain/test accuracy as utility checks. CR and ED were reserved for final evaluation. Table 6 lists the frozen update budgets and principal hyperparameters.

Unless otherwise stated in the table, SCRUB, SalUn, SFR-on, and Sharp MinMax use SGD with momentum 0.9 and weight decay 5×10^{-4}; UGradSL, clipped GA/FedGA, EWCSGA, and FedOSD use zero momentum and weight decay. FedFT uses momentum 0.9 and zero weight decay. SFR-on additionally uses an ascent-gradient cap of 7, Fisher batch size 128, adaptive coefficient 0.5, saliency threshold 1, saliency epsilon 10^{-15}, and fast/slow coefficient 1. Sharp MinMax uses sharpness coefficient 1 and adaptive perturbations.

FUPGA estimates its projection radius from ten random models (squared-distance divisor 3); its reported endpoint includes one full-participation FedAvg continuation round using 3,200 retained examples per physical client, batch size 128, LR 0.01, momentum 0.9, and zero weight decay. EWCSGA estimates its Original-anchored Fisher matrix from 512 retained examples at the requesting client, with batch size 1. FedOSD uses a singular-value cutoff of 10^{-6}; we report its unlearning-stage endpoint without its native projected recovery. For methods defined with a dedicated forgetting client, the 100 requested samples form a virtual forget client while the remaining samples of physical client 0 retain their original federation assignment.

Table 8: Frozen CIFAR-10 ReCo settings used in both training environments.

Parameter	Value
Reserved/active nonmembers	$R_{\max} = 960$, $M = 240$ (80 per A/B/C split)
Neighbor candidates	$\{3, 5, 7, 11, 15, 23, 31\}$; select on A-to-B median MAE
Quantile rank clipping	$[0.02, 0.98]$
Score/control scale	Global $1.4826\,\mathrm{MAD}$, floor 10^{-3}
Control temperature T; potential tolerance	0.01; 10^{-3}
Active tolerance; hysteresis; target offset	0; 0; 0
Erasure SGD	LR 0.005, batch 128, momentum/WD $0/0$
Maximum steps; minimum calls; stop patience	100; 1; 3 complete-request checks
Difficulty and controller evaluation batch	256
Secant damping; worsening shrink	0.7; 0.5
Step-scale range; growth cap	$[0.02, 8]$; 2
Per-step relative parameter drift cap	3.5×10^{-4}
Backtracking	Factor 0.5, at most 8 reductions, minimum 2^{-8}
Acceptance tolerance; required improvement	10^{-8}; 0

A.4 ReCo Hyperparameters

The centralized and federated runs use the same frozen ReCo hyperparameters listed in Table 8. The estimator, violation potential, and stopping rule are specified in Appendix C.

A.5 Evaluation and Retained-Data Training

Membership-inference evaluation. We use online LiRA with 16 shadow models in each training setting: eight IN and eight OUT models for every queried example. The attack signal is the true-label log-odds s_i defined in the main paper. Gaussian distributions are fitted to the IN and OUT signals, and the corresponding log-likelihood ratio is evaluated by ROC-AUC. The positive population comprises the 100 deleted training examples. Negatives consist of 100 class-matched held-out validation examples centrally or 2,000 test examples federally, excluding ReCo's reserved nonmembers. Audit banks are shared by all methods within each request. We additionally report RMIA and a class-matched proxy MIA in the supplementary tables; the latter is used only for development requests when adapting baselines.

BLORO evaluation. The main experiments use $K = 5$ retraining references and nominal coverage $1 - \alpha = 0.9$. Thresholds for a query request are calibrated from the other two deletion requests within the same training world. Retraining references are used exclusively for offline evaluation, not for unlearning updates or checkpoint selection. Conventional metrics for Retrain/FedRetrain are averaged across the five retraining models per request, whereas CR and ED use BLORO's held-out-retrain-as-query construction. The resulting reference CR is therefore expected to be near the nominal coverage, not one. Section B gives the calibration equations and sensitivity analysis.

Subsequent retained-data training. The persistence experiment continues training from each frozen CIFAR-10 unlearning endpoint with retained data only and no additional unlearning controller. Centralized continuation uses 400 SGD steps, batch size 128, LR 0.005, momentum 0.9, and zero weight decay. Federated continuation uses 20 full-participation FedAvg rounds with one local epoch on 512 retained examples per physical client per round, batch size 128, LR 0.005, momentum 0.9, and zero weight decay; sampling cycles without replacement. Both protocols process 51,200 retained-example exposures. Retrain/FedRetrain remain unchanged as reference rows, and BLORO retains the original reference models and calibrated thresholds.

A.6 Supplementary Evaluation Metrics

Tables 9 and 10 complement the main results with retained accuracy, auxiliary privacy audits, and the directional incompatibility rates defined in Appendix B.2. Each cell reports the mean and sample standard deviation across six requests; retraining set-level metrics are first averaged over five reference models per request.

Table 9: Additional centralized CIFAR-10 metrics. Mean ± sample standard deviation over six requests; UR and OR denote under- and over-forgetting rates.

Method	Retain Acc.	Proxy MIA AUC	RMIA AUC	UR	OR
Original	1.000 ± 0.000	0.650 ± 0.033	0.723 ± 0.017	0.193 ± 0.027	0.026 ± 0.016
Retrain	1.000 ± 0.000	0.494 ± 0.026	0.521 ± 0.033	0.050 ± 0.012	0.052 ± 0.005
FT	1.000 ± 0.000	0.645 ± 0.032	0.739 ± 0.015	0.113 ± 0.020	0.175 ± 0.044
GA (clip)	0.995 ± 0.002	0.632 ± 0.031	0.709 ± 0.016	0.164 ± 0.017	0.061 ± 0.033
SCRUB	0.999 ± 0.001	0.512 ± 0.038	0.567 ± 0.030	0.068 ± 0.029	0.223 ± 0.035
SalUn	1.000 ± 0.000	0.397 ± 0.029	0.449 ± 0.028	0.037 ± 0.014	0.344 ± 0.037
SFR-on	0.969 ± 0.051	0.293 ± 0.172	0.257 ± 0.161	0.041 ± 0.039	0.649 ± 0.252
UGradSL	0.968 ± 0.017	0.502 ± 0.030	0.546 ± 0.021	0.066 ± 0.028	0.281 ± 0.065
Sharp MinMax	0.978 ± 0.006	0.401 ± 0.034	0.388 ± 0.018	0.007 ± 0.006	0.537 ± 0.066
ReCo	0.994 ± 0.005	0.478 ± 0.041	0.518 ± 0.044	0.030 ± 0.009	0.165 ± 0.059

Table 10: Additional federated CIFAR-10 metrics. Mean ± sample standard deviation over six requests; UR and OR denote under- and over-forgetting rates.

Method	Retain Acc.	Proxy MIA AUC	RMIA AUC	UR	OR
Original	1.000 ± 0.000	0.657 ± 0.022	0.719 ± 0.027	0.266 ± 0.045	0.023 ± 0.013
FedRetrain	1.000 ± 0.000	0.513 ± 0.045	0.499 ± 0.020	0.051 ± 0.013	0.050 ± 0.012
FedFT	1.000 ± 0.000	0.657 ± 0.022	0.718 ± 0.027	0.267 ± 0.047	0.023 ± 0.013
FedGA (clip)	1.000 ± 0.000	0.646 ± 0.026	0.697 ± 0.024	0.249 ± 0.048	0.033 ± 0.018
FUPGA	0.796 ± 0.180	0.550 ± 0.086	0.553 ± 0.114	0.088 ± 0.114	0.360 ± 0.220
EWCSGA	0.551 ± 0.491	0.575 ± 0.090	0.795 ± 0.127	0.162 ± 0.095	0.480 ± 0.476
FedOSD (unlearning stage)	0.998 ± 0.002	0.574 ± 0.042	0.543 ± 0.037	0.170 ± 0.056	0.147 ± 0.039
ReCo	0.985 ± 0.013	0.479 ± 0.051	0.464 ± 0.041	0.034 ± 0.017	0.218 ± 0.031

In the fixed federated EWCSGA configuration, one request produces extreme score values. Its untrimmed ED mean (sample SD) over six requests is 1.29×10^{17} (3.16×10^{17}).

B BLORO: Calibration and Supplementary Analysis

B.1 Request-Disjoint Threshold Calibration

Let q index a deletion request in one training world and let z^R_{qik} denote the standardized deviation of held-out retraining model k on deleted sample i, calculated against the other $K-1$ references as in the main paper. For a query request q, construct the calibration pool from the other requests in that world:

$$\mathcal{Z}_R^{(-q)} = \big\{ z^R_{q'ik} : q' \neq q,\ q' \text{ in the same world} \big\}. \tag{B.1}$$

The lower and upper compatibility thresholds are

$$\tau_L^{(-q)} = Q_{\alpha/2}\big(\mathcal{Z}_R^{(-q)}\big), \qquad \tau_U^{(-q)} = Q_{1-\alpha/2}\big(\mathcal{Z}_R^{(-q)}\big). \tag{B.2}$$

All methods and retrain-as-query comparisons for request q use these same thresholds. The calibration request pool is disjoint from the evaluated request, and all reported rates are aggregated over deleted-sample/reference-fold pairs.

B.2 Directional Incompatibility Rates

Because larger true-label log-odds indicate stronger support for the original label, scores above the upper threshold indicate under-forgetting and scores below the lower threshold indicate over-

Table 11: Centralized CIFAR-10 held-out retraining coverage for target request 2230, calibrated with request 2231 in world 2027. Median [25th, 75th percentile] over 100 reference-subset pairs; these repetitions are not new deletion requests.

K	Held-out target retrains	Direct min–max	BLORO
3	17	0.499 [0.478, 0.518]	0.895 [0.883, 0.909]
5	15	0.667 [0.643, 0.688]	0.901 [0.886, 0.915]
10	10	0.818 [0.800, 0.831]	0.906 [0.889, 0.919]
15	5	0.877 [0.865, 0.892]	0.906 [0.890, 0.927]

Table 12: Centralized CIFAR-10 method assessment for the same target/calibration requests. Entries are median CR and ED over 100 subsets for $K < 20$, and the single full-reference result for $K = 20$.

K	ReCo CR	ReCo ED	SalUn CR	SalUn ED	Sharp MinMax CR	Sharp MinMax ED
3	0.827	6.751	0.763	9.408	0.697	13.877
5	0.706	0.675	0.561	1.371	0.435	2.134
10	0.622	0.467	0.472	1.074	0.355	1.693
15	0.615	0.443	0.438	1.016	0.324	1.623
20	0.599	0.437	0.403	1.017	0.315	1.623

forgetting. These two directions are measured by

$$\mathrm{UR} = \frac{1}{|\mathcal{D}_f|K} \sum_{i,k} \mathbf{1}\big[z_{ik}^U > \tau_U\big], \tag{B.3}$$

$$\mathrm{OR} = \frac{1}{|\mathcal{D}_f|K} \sum_{i,k} \mathbf{1}\big[z_{ik}^U < \tau_L\big]. \tag{B.4}$$

Together with the Compatibility Rate, they satisfy $\mathrm{CR} + \mathrm{UR} + \mathrm{OR} = 1$. For sample-level visualizations, the corresponding rates can be obtained by averaging the same indicators over k for each deleted sample.

B.3 Numerical Details and Reference-Count Sensitivity

BLORO computes true-label log-odds as the stable logit margin $z_y - \mathrm{logsumexp}_{j \neq y} z_j$ and evaluates the standardized scores in float64. It uses the sample standard deviation of the $K - 1$ reference scores (thus $K - 2$ degrees of freedom) and fixed $\epsilon = 10^{-8}$ in the denominator. Quantiles use linear interpolation.

To study reference-count sensitivity, we additionally use two centralized 20-retraining-reference assets from world 2027: request 2230 is the evaluation request and request 2231 provides out-of-request calibration. These assets are used only in the sensitivity analysis; the main tables always use their first five references. For each $K \in \{3, 5, 10, 15\}$, we independently draw 100 pairs of reference subsets from the two 20-model pools, calibrating new thresholds for every pair. The unused target retrains act as held-out queries for the coverage analysis. The $K = 20$ result serves only as a full-reference method-assessment anchor. Subset sampling uses seed $20270923 + 1009K$.

Table 11 gives the numerical coverage results underlying the main calibration comparison. The corresponding method-level scores are shown in Table 12. BLORO maintains near-nominal held-out coverage across the tested reference counts, while absolute CR and ED values can vary with reference resolution.

The empirical min–max interval is included only as a reference-count comparison: its coverage changes with the number of retraining draws because its endpoints are sample-size-dependent order statistics. BLORO instead calibrates against genuine retrain-to-retrain deviations under a matched number of reference models.

B.4 Evaluation Procedure

Given a realized unlearned model, a deletion set, K exact-retraining references, and thresholds calibrated from other requests, BLORO proceeds as follows:

1. For each reference k, use the other $K-1$ retrained models to compute each deleted sample's reference mean and standard deviation.
2. Standardize both the held-out retrained model and the realized unlearned model against that same reference fold.
3. Compare the unlearned standardized scores against the calibrated thresholds and aggregate the sample–fold indicators and deviations to obtain CR and ED. UR and OR may be reported as directional diagnostics.

C ReCo Implementation and Proxy-Target Analysis

C.1 Deterministic Multi-View Difficulty

The image-classification implementation constructs seven deterministic views: the original image, its horizontal flip, four corner crops, and one center crop. The crops are taken after reflection padding by two pixels on every side; the corner origins are $(0,0),(0,4),(4,0),(4,4)$ and the center origin is $(2,2)$. Each view is evaluated by the original model on clean, normalized images. With $s_i^{(v)}$ denoting its true-label log-odds under view v, the descriptor is

$$\begin{aligned} \widetilde{s}_i &= Q_{0.5}\big(\{s_i^{(v)} : v \in \mathcal{V}\}\big), \\ d_i &= Q_{0.2}\big(\{s_i^{(v)} : v \in \mathcal{V}\}\big) - Q_{0.5}\big(\{|s_i^{(v)} - \widetilde{s}_i| : v \in \mathcal{V}\}\big). \end{aligned} \tag{C.1}$$

The view-level median absolute deviation is used without Gaussian rescaling. All difficulty measurements are completed before unlearning and remain fixed.

C.2 Conditional Proxy Targets and Their Validation

Conditional-transfer motivation. Let $F_N(t \mid d)$ denote the conditional score CDF of held-out nonmembers under the original model, and let $F_R(t \mid d)$ denote the corresponding score CDF of deleted samples under stochastic exact retraining, pooling samples at a common original-model difficulty d. ReCo uses the conditional nonmember median as a proxy for the central location of the latter. This is motivated by a task-dependent conditional-transfer assumption, which may be written as

$$\sup_t \big|F_N(t \mid d) - F_R(t \mid d)\big| \le \delta(d). \tag{C.2}$$

For a proxy $c(d)$ whose nonmember conditional CDF value differs from $1/2$ by at most $\eta(d)$, the triangle inequality immediately gives $|F_R(c(d) \mid d) - 1/2| \le \delta(d) + \eta(d)$. We examine the practical usefulness of this conditional-location proxy against held-out retraining scores below.

Target estimation. The active nonmember pools $\mathcal{N}_A$ and $\mathcal{N}_B$ are used for neighborhood selection and fitting; $\mathcal{N}_C$ is held out for residual calibration. To select the monotone orientation, compute the Spearman correlation ρ_A between difficulty and true-label score on $\mathcal{N}_A$, and define $o = +1$ when $\rho_A \ge 0$ and $o = -1$ otherwise. Neighbor search uses the oriented difficulty $u = od$ and robustly normalized distances. With h_i the distance from query i to its k-th nearest neighbor, each reference receives tricube weight

$$w_{ij} = \max\big\{(1 - r_{ij}^3)^3, 10^{-6}\big\}, \qquad r_{ij} = \min\big\{\mathrm{dist}_{ij}\,/h_i, 1\big\}. \tag{C.3}$$

Weighted empirical quantiles are fitted on $\mathcal{N}_A$; the number of neighbors is selected by median prediction MAE on $\mathcal{N}_B$. The estimator is then refitted on $\mathcal{N}_A \cup \mathcal{N}_B$, and weighted isotonic regression followed by linear interpolation produces the monotone conditional-median curve $\widehat{Q}_{0.5}(s \mid d)$.

The held-out pool $\mathcal{N}_C$ supplies the residual-location correction

$$\Delta = \mathrm{median}_{j \in \mathcal{N}_C}\big[s_j - \widehat{Q}_{0.5}(s \mid d_j)\big], \qquad c_i = \widehat{Q}_{0.5}(s \mid d_i) + \Delta. \tag{C.4}$$

Table 13: CIFAR-10 proxy error against each sample's five-reference retraining median. Mean ± request-level SD of MAE in log-odds units; six requests per setting.

Setting	Unconditional	Conditional, no shift	Saved ReCo target
Centralized	7.645 ± 0.675	4.211 ± 0.502	4.256 ± 0.617
Federated	5.771 ± 0.416	3.278 ± 0.250	3.298 ± 0.240

A common, fixed score scale for each request is estimated from the fitting nonmembers:

$$a = \max\left\{1.4826 \ \mathrm{median}_j \left|s_j - \mathrm{median}_r(s_r)\right|, 10^{-3}\right\}, \quad j, r \in \mathcal{N}_A \cup \mathcal{N}_B. \tag{C.5}$$

Both the targets and scale are frozen before the controller begins updating the model. The active controller uses the conditional median as its target.

Offline target validation. We compare the saved ReCo targets with the median score of each deleted sample under its five exact-retraining references. An unconditional baseline uses the median of the same 160 fitting nonmembers; the unshifted conditional predictor isolates the contribution of difficulty conditioning. All proxy predictors are determined without access to retraining observations. Table 13 summarizes their samplewise target errors over the six formal requests in each setting.

Difficulty conditioning substantially reduces proxy error relative to the unconditional target in both settings. The calibrated and unshifted conditional estimates have similar MAE, while the final ReCo configuration uses the frozen split-C residual correction specified above.

C.3 Violation Potential and Adaptive Forgetting

At each step, ReCo computes normalized target violations $m_i = (s_i - c_i)/a$ and activates samples with $m_i > 0$, as defined in Eq. (10). To combine typical and large violations without diluting the latter, the potential groups deleted samples by class. For group g, define

$$A_g = \frac{1}{|g|}\sum_{i \in g}[m_i]_+, \qquad B_g = \max_{i \in g} m_i, \tag{C.6}$$

and $D_g = A_g - \varepsilon$, $T_g = [B_g + A_g]_+ - \varepsilon$. With

$$\mathrm{Agg}(v_1, \ldots, v_n) = \max_j v_j + \frac{1}{n}\sum_{j=1}^{n}[v_j]_+, \tag{C.7}$$

the request-level potential is

$$E_g = \mathrm{Agg}(D_g, T_g), \qquad \Phi = \mathrm{Agg}\big(\{E_g\}_g\big). \tag{C.8}$$

Here A_g captures mean positive violation and B_g retains the largest violation in each class. ReCo obtains the score-space priorities $q_i = \partial\Phi/\partial s_i$, gates inactive samples with $b_i = \mathbf{1}[m_i > 0]$, and applies the request-level scale $g_\Phi = \sigma(\Phi/T)$. The resulting weights $\widetilde{q}_i = g_\Phi b_i q_i$ are detached before being applied to the erasure objective in Eq. (12). Class labels enter this prioritization through the grouping and the true-label score; the conditional target estimator is fitted without class-wise partitions.

C.4 Adaptive Steps and Automatic Stopping

The base erasure learning rate is multiplied by an adaptive scale combining a secant estimate of recent progress, backoff after worsening potential, and feedback from parameter drift. Changes in this scale are capped to avoid abrupt growth. For controller-binding parameters θ_{bind}, each candidate step obeys

$$\|\Delta\theta_t\|_2 \le \delta\|\theta_{\mathrm{bind}}\|_2. \tag{C.9}$$

A candidate update exceeding this relative-drift budget is rescaled. ReCo then evaluates progressively smaller candidate steps on the complete forget set and accepts the first satisfying

$$\Phi(\theta_t + \alpha\Delta\theta_t) \leq \Phi(\theta_t) + \epsilon_{\mathrm{bt}}, \qquad \alpha \in \{1, 1/2, 1/4, \ldots\}. \tag{C.10}$$

If none satisfies the acceptance condition, the update is reverted. The controller checks the complete forget set after every accepted update and stops once all samples satisfy $s_i(\theta) \leq c_i$ for the configured number of consecutive checks, or at its maximum step budget. The targets remain fixed throughout; any subsequent retained-data training proceeds without the controller.