COFORGET: COMPOSING HETEROGENEOUS UNLEARNING REQUESTS TO ACTIVATE DORMANT BACKDOORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unlearning-activated backdoor attacks keep a backdoor dormant after training and activate it later through an unlearning request. Requests composed entirely of camouflage samples can induce this transition but may also reduce benign accuracy. Since clean samples can influence backdoor behavior, incorporating them alongside camouflage samples provides an additional way to improve this balance. To this end, we introduce CoForget, a request-composition attack that combines a small fixed camouflage component with deliberately selected clean samples. CoForget first calibrates a universal trigger with a logit gap objective to promote dormancy without excessively suppressing the target response. It then greedily selects a clean subset whose cumulative gradient aligns with an activation direction estimated on a local shadow model. Neither trigger nor request construction requires access to the victim model or knowledge of the provider's unlearning algorithm. Experiments across three image classification datasets and three approximate unlearning algorithms show low attack success before unlearning and high success afterward. In the evaluated clean component ablations, selected samples improve mean benign accuracy over camouflage-only requests and retain high activation. Comparisons at matched total request sizes further show strong activation while using substantially fewer camouflage samples than the evaluated baselines. These findings motivate considering request composition when assessing unlearning security.

1 INTRODUCTION

Machine unlearning supports data revocation by eliminating the influence of specified training samples (Cao & Yang, 2015; Ginart et al., 2019). However, attackers can exploit the unlearning request to manipulate model behavior (Qian et al., 2023; Hu et al., 2024; Huang et al., 2025). Unlearning-activated backdoor attacks keep malicious behavior dormant after training and activate it when a designated request is processed (Zhang et al., 2023; Huang et al., 2024; Alam et al., 2025). Unlike conventional backdoors, whose triggered behavior is directly manifest in the trained model (Gu et al., 2017; Chen et al., 2017; Zeng et al., 2023), these attacks deliberately suppress it before unlearning. Attackers seek to achieve this transition while limiting accuracy loss on clean inputs.

Existing unlearning-activated backdoor attacks follow two routes. One established mechanism builds on the removable camouflage introduced by Hidden Poison (Di et al., 2023). In these attacks, poisoned training samples establish an association between the trigger and the target class. Camouflage samples provide a competing training signal that suppresses backdoor activation. Unlearning camouflage samples removes this suppression, thereby activating the dormant backdoor (Zhang et al., 2023; Huang et al., 2024; Alam et al., 2025; Zhao et al., 2026). Other work explores unlearning clean samples to activate or amplify backdoor behavior (Liu et al., 2024; Arazzi et al., 2025).

Although camouflage-only requests can induce the desired transition, the resulting updates may impair model utility. Since unlearning clean samples can also affect backdoor behavior, we consider incorporating clean samples into the request to shape the aggregate update. However, arbitrary clean subsets do not ensure activation. This raises a question: can deliberately selected clean samples improve the balance between activation and benign utility in a heterogeneous unlearning request?

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 answer this question with CoForget, a request-composition attack. CoForget holds a small camouflage component fixed and selects a clean subset to accompany it. The construction uses local models and requires neither victim queries nor knowledge of the deployed unlearning algorithm. CoForget first calibrates a universal trigger using a soft penalty on the logit gap. The calibration promotes dormancy before unlearning while retaining a tendency toward the target class. Using the calibrated trigger, CoForget estimates an activation direction on a local shadow model. At each step, the greedy selector adds the candidate that maximizes the alignment of the updated cumulative gradient with this direction. In the clean component ablations, requests containing selected clean samples achieve higher mean benign accuracy (BA) than camouflage-only requests while maintaining a high attack success rate (ASR) after unlearning.

Our contributions are summarized as follows:

- We study the joint unlearning of camouflage and unmodified clean samples, showing that clean sample selection can improve the balance between backdoor activation and benign accuracy in the evaluated settings.
- We develop logit gap calibration to control trigger strength and a greedy selector for clean samples. The selector aligns their cumulative gradient with an activation direction estimated on a local shadow model.
- Evaluations across three datasets and three approximate unlearning algorithms document the transition from dormancy to activation. Ablations show that selected clean samples improve mean BA over camouflage-only requests while retaining high ASR. With the total request budget matched, CoForget maintains strong activation with substantially fewer camouflage samples than the evaluated baselines.

2 Related Work

Machine unlearning. Machine unlearning aims to eliminate the influence of specified training samples while preserving model utility on the retained data (Cao & Yang, 2015; Ginart et al., 2019). Retraining from scratch on the retained data provides a reference solution, but it is computationally expensive. Frameworks such as SISA reduce this cost by localizing the influence of training samples and retraining only the affected model components (Bourtoule et al., 2021). Approximate unlearning methods instead update the parameters of an existing model to approximate the outcome of retraining. Representative approaches include analytical parameter corrections (Guo et al., 2020; Izzo et al., 2021; Sekhari et al., 2021), weight scrubbing (Golatkar et al., 2020), and gradient-based updates (Neel et al., 2021). More recent work explores teacher–student objectives (Kurmanji et al., 2023), fine-tuning with adversarial examples (Ebrahimpour-Boroojeny et al., 2025), and sharpness-aware optimization (Tang & Khanna, 2026) for unlearning.

Unlearning-activated backdoors. Unlearning-activated backdoor attacks manipulate the training and forget sets to induce backdoor activation. Hidden Poison (Di et al., 2023) demonstrated that camouflage samples can conceal targeted poisoning until their removal. BAU (Zhang et al., 2023) implants a backdoor via poisoned training samples and suppresses its activation with mitigation samples. UBA-Inf (Huang et al., 2024) synthesizes camouflage samples via influence-driven perturbations, whereas ReVeil (Alam et al., 2025) introduces noisy camouflage without requiring auxiliary data or direct model access. Zhao et al. (2026) jointly optimize a trigger generator and a label-consistent camouflage generator via a bilevel objective. Despite differences in sample construction, these approaches focus request design on specially crafted camouflage or mitigation samples.

Another line of work investigates backdoor activation through the unlearning of clean training samples. Under approximate unlearning, BAVU (Liu et al., 2024) selects a subset of attacker data through continuous relaxation to construct an unlearning request. UNCLEAN (Arazzi et al., 2025) embeds a trigger across classes while preserving the original labels. Unlearning clean non-target samples weakens competing class associations, which in turn amplifies the backdoor. Prior work indicates that unlearning either camouflage samples or clean samples can influence backdoor activation. CoForget constructs heterogeneous unlearning requests that combine camouflage samples with a selected clean subset.

3 Problem Setup and Threat Model

3.1 Problem Formulation

Let $\mathcal{D}$ denote the training set collected by the service provider, with inputs in $\mathcal{X}$ and labels in $\mathcal{Y} = \{1, \ldots, K\}$. The provider trains a classifier f_θ on $\mathcal{D}$ and processes a request to unlearn $\mathcal{D}_f \subseteq \mathcal{D}$ using an approximate unlearning algorithm, producing the updated model f_{θ^-}. The attacker constructs a heterogeneous unlearning request whose forget set contains two types of samples:

$$\mathcal{D}_f = \mathcal{D}_{\text{cam}} \cup \mathcal{D}^\star_{\text{cln}}. \tag{1}$$

We denote the attack target class by $t \in \mathcal{Y}$ and the trigger transformation by $T_\delta : \mathcal{X} \to \mathcal{X}$. Given samples from the target and non-target classes, denoted by $\mathcal{S}_{\text{poi}}$ and $\mathcal{S}_{\text{cam}}$, respectively, the attacker constructs

$$\begin{aligned} \mathcal{D}_{\text{poi}} &= \left\{ \big(T_\delta(x), t\big) \,\middle|\, (x,t) \in \mathcal{S}_{\text{poi}} \right\}, \\ \mathcal{D}_{\text{cam}} &= \left\{ \big(T_\delta(x), y\big) \,\middle|\, (x,y) \in \mathcal{S}_{\text{cam}},\ y \neq t \right\}. \end{aligned} \tag{2}$$

Both components preserve the original labels of their base samples. The poison samples from the target class are included in the training set, $\mathcal{D}_{\text{poi}} \subseteq \mathcal{D}$. The clean component $\mathcal{D}^\star_{\text{cln}}$ contains selected clean samples. Section 4 describes the trigger construction and the proposed clean sample selection method.

3.2 Threat Model

Adversarial Goal. The attacker aims to keep the backdoor dormant before unlearning and activate it afterward. This objective corresponds to low ASR on triggered non-target test samples before unlearning and high ASR afterward. The attacker also seeks to limit the decrease in BA on the clean test set.

Adversarial Capabilities. The attacker acts as a legitimate data contributor and locally holds a set of unmodified downstream samples, denoted by $\mathcal{D}_A = \mathcal{D}_A^t \cup \mathcal{D}_A^{\neg t}$. Here, $\mathcal{D}_A^t$ and $\mathcal{D}_A^{\neg t}$ denote the target and non-target subsets, respectively. The sets of base samples used to construct poison and camouflage samples satisfy $\mathcal{S}_{\text{poi}} \subseteq \mathcal{D}_A^t$ and $\mathcal{S}_{\text{cam}} \subseteq \mathcal{D}_A^{\neg t}$. The attacker also has access to a public auxiliary dataset $\mathcal{D}_s$ from a related domain. The attacker uses only local models to construct the trigger and unlearning request, without accessing or querying the victim model. This process does not require knowing which approximate unlearning algorithm the provider uses.

During data collection, the attacker constructs $\widetilde{\mathcal{D}}_A$ from $\mathcal{D}_A$ and submits it to the provider for training. This submitted dataset contains target poison samples, camouflage samples and unmodified clean samples. All samples requested for unlearning belong to this submitted contribution, satisfying $\mathcal{D}_f \subseteq \widetilde{\mathcal{D}}_A \subseteq \mathcal{D}$. After deployment, the attacker submits the unlearning request for backdoor activation. The provider retains full control over training and unlearning, and the attacker cannot directly modify the victim model.

Deployment Scope. We consider services where users contribute labeled data for model training. The attacker's unlearning requests are limited to samples that it contributed to the victim's training set. The provider retains the authority to inspect training data, screen unlearning requests, and repair the model.

4 Methodology

As shown in Figure 1, CoForget constructs a heterogeneous unlearning request in two stages. It first calibrates a universal trigger and uses it to construct target poisons and camouflage samples. With the camouflage component fixed, it then selects clean samples whose cumulative gradient aligns with an estimated activation direction.

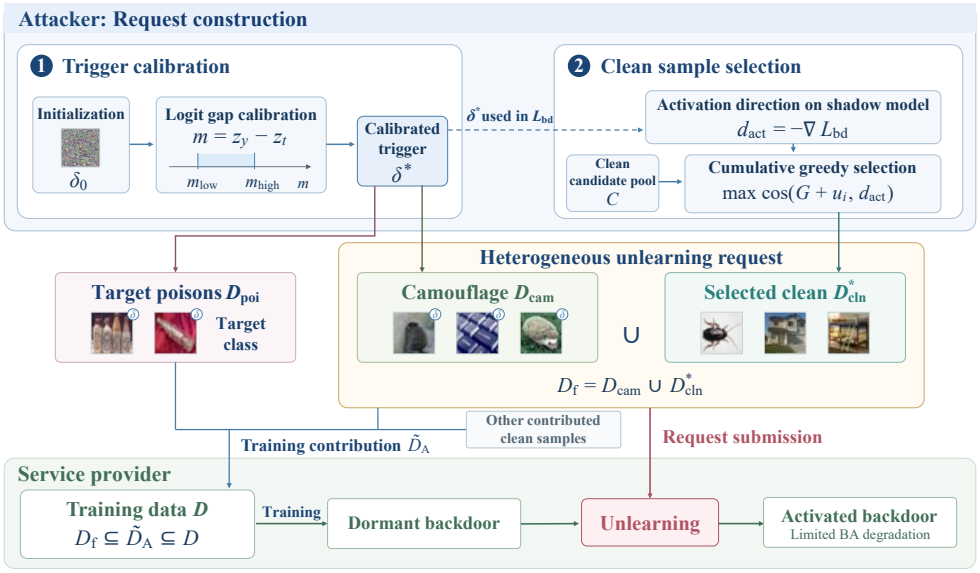


Figure 1: Overview of CoForget. The attacker uses a calibrated trigger to construct target poisons and camouflage samples, and greedily selects a clean subset through cumulative gradient alignment. The camouflage and selected clean samples form a heterogeneous unlearning request. Processing this request after deployment activates the dormant backdoor while preserving benign accuracy.

4.1 Calibrated Trigger Construction

Trigger initialization. Let δ denote a universal additive perturbation, with the trigger transformation defined as

$$T_\delta(x) = \mathrm{clip}_{\mathcal{X}}(x + \delta), \qquad \|\delta\|_\infty \le \epsilon, \tag{3}$$

where $\mathrm{clip}_{\mathcal{X}}$ restricts the perturbed input to the valid range and ϵ specifies the perturbation budget.

Inspired by the inward trigger design of Narcissus (Zeng et al., 2023), trigger initialization uses a locally trained source model. Let $\mathcal{Y}_s$ denote the label space of $\mathcal{D}_s$, with $K_s = |\mathcal{Y}_s|$. Samples in $\mathcal{D}_s$ retain their original labels, whereas all target class samples in $\mathcal{D}_A^t$ are assigned an auxiliary label c_{aux}. A source model $f_{\mathrm{src}} : \mathcal{X} \to \mathbb{R}^{K_s+1}$ is then trained on these data, with its first K_s outputs corresponding to $\mathcal{Y}_s$ and the additional output corresponding to c_{aux}.

Let $\ell(f(x), y)$ denote the cross-entropy loss for model output $f(x)$ and label y. The initial trigger is obtained by

$$\delta_0 = \arg\min_{\|\delta\|_\infty \le \epsilon} \frac{1}{|\mathcal{D}_A^t|} \sum_{(x,t)\in\mathcal{D}_A^t} \ell(f_{\mathrm{src}}(T_\delta(x)), c_{\mathrm{aux}})\,. \tag{4}$$

This objective encourages the trigger to encode a direction toward the auxiliary class associated with the downstream target class. The resulting δ_0 serves as the starting point and reference for downstream logit gap calibration.

Logit gap calibration. The source model f_{src} is adapted to the downstream task by replacing its output head and fine-tuning on $\mathcal{D}_A$. The resulting K-class shadow model is denoted by $f_{\mathrm{sh}}(\cdot;\theta_{\mathrm{sh}}) : \mathcal{X} \to \mathbb{R}^K$. The trigger is optimized from δ_0 with the shadow model fixed.

For a non-target sample $(x, y) \in \mathcal{D}_A^{\neg t}$, the logit gap on its triggered input is defined as

$$m(x, y; \delta) = f_{\mathrm{sh},y}(T_\delta(x)) - f_{\mathrm{sh},t}(T_\delta(x)). \tag{5}$$

A positive calibration interval $0 < m_{\mathrm{low}} < m_{\mathrm{high}}$ specifies the desired logit gap range, with deviations penalized by

$$\mathcal{L}_{\mathrm{cal}}(\delta) = \frac{1}{|\mathcal{D}_A^{\neg t}|} \sum_{(x,y)\in\mathcal{D}_A^{\neg t}} \left[\mathrm{ReLU}(m_{\mathrm{low}} - m(x, y; \delta)) + \mathrm{ReLU}(m(x, y; \delta) - m_{\mathrm{high}})\right]. \tag{6}$$

The lower term encourages an advantage for the true class, while the upper term discourages an overly weak target response relative to the true class. This calibration balances dormancy before unlearning with the signal toward the target class.

The trigger is optimized using the logit gap loss, with regularization toward δ_0:

$$\delta^\star = \arg\min_{\|\delta\|_\infty \le \epsilon} \lambda_{\mathrm{cal}} \mathcal{L}_{\mathrm{cal}}(\delta) + \lambda_{\mathrm{anc}} \|\delta - \delta_0\|_2^2. \tag{7}$$

Accordingly, the calibrated trigger $\delta^\star$ is used in Equation 2 to construct $\mathcal{D}_{\mathrm{poi}}$ and $\mathcal{D}_{\mathrm{cam}}$.

4.2 Gradient Alignment for Clean Sample Selection

CoForget constructs the clean component by aligning its cumulative training gradient with an activation direction estimated on the shadow model.

Activation gradient. Let $\mathcal{D}_{\mathrm{nt}} \subseteq \mathcal{D}_A^{\neg t}$ denote a set of non-target samples drawn from the attacker data, and let ϕ_{sh} denote the subvector of θ_{sh} used for gradient comparison. The activation direction is estimated from the target-label loss on triggered non-target samples:

$$\begin{aligned} \mathcal{L}_{\mathrm{bd}} &= \frac{1}{|\mathcal{D}_{\mathrm{nt}}|} \sum_{(x,y)\in\mathcal{D}_{\mathrm{nt}}} \ell(f_{\mathrm{sh}}(T_{\delta^\star}(x); \theta_{\mathrm{sh}}), t)\,, \\ g_{\mathrm{bd}} &= \nabla_{\phi_{\mathrm{sh}}} \mathcal{L}_{\mathrm{bd}}, \qquad d_{\mathrm{act}} = -g_{\mathrm{bd}}. \end{aligned} \tag{8}$$

Since decreasing $\mathcal{L}_{\mathrm{bd}}$ promotes predictions of the target class on triggered non-target samples, its negative gradient $d_{\mathrm{act}} = -g_{\mathrm{bd}}$ defines the local activation direction used to align clean samples.

Selection objective. Let $\mathcal{C} = \{z_i = (x_i, y_i)\}_{i=1}^M \subseteq \mathcal{D}_A^{\neg t} \setminus \mathcal{S}_{\mathrm{cam}}$ denote the clean candidate pool. Each candidate gradient is computed using the original label:

$$u_i = \nabla_{\phi_{\mathrm{sh}}} \ell(f_{\mathrm{sh}}(x_i; \theta_{\mathrm{sh}}), y_i)\,. \tag{9}$$

Using these gradients, a coarse filter retains candidates with the largest individual alignment $\cos(u_i, d_{\mathrm{act}})$. Let $\mathcal{J} \subseteq \{1, \dots, M\}$ denote the indices of the retained candidates. For $\mathcal{I} \subseteq \mathcal{J}$, define

$$G(\mathcal{I}) = \sum_{i\in\mathcal{I}} u_i,\ n_c(\mathcal{I}) = \sum_{i\in\mathcal{I}} \mathbf{1}\{y_i = c\}, \tag{10}$$

where $G(\mathcal{I})$ is the cumulative shadow gradient for subset $\mathcal{I}$, and $n_c(\mathcal{I})$ counts the selected samples from class c.

To score clean subsets, we use a local proxy based on gradient ascent on their training loss. With gradients evaluated at the fixed shadow parameters, the hypothetical proxy step is $\Delta\phi_{\mathrm{proxy}} = \eta G(\mathcal{I})$ with $\eta > 0$. For a sufficiently small proxy step, a first-order expansion gives

$$\Delta\mathcal{L}_{\mathrm{bd}}^{\mathrm{proxy}} \approx \eta \langle g_{\mathrm{bd}}, G(\mathcal{I})\rangle = -\eta \langle d_{\mathrm{act}}, G(\mathcal{I})\rangle\,.$$

Under this proxy, positive alignment corresponds to a first-order decrease in the target-label loss on triggered non-target samples. The selector uses cosine similarity as a normalized alignment score to select k_{cln} clean samples subject to a per-class cap q:

$$\begin{aligned} &\max_{\mathcal{I}\subseteq\mathcal{J}} \quad \cos(G(\mathcal{I}), d_{\mathrm{act}}) \\ &\text{s.t.} \quad |\mathcal{I}| = k_{\mathrm{cln}},\ n_c(\mathcal{I}) \le q,\ \forall c \in \mathcal{Y} \setminus \{t\}. \end{aligned} \tag{11}$$

Appendix B provides a local analysis of this objective and its cumulative evaluation.

Greedy selection. The cumulative alignment objective is optimized using a greedy procedure, avoiding exhaustive search over candidate subsets. A per-class cap limits concentration in a few classes. We deliberately initialize $\mathcal{I}_0 = \emptyset$ and $G_0 = 0$, so that the selector directly optimizes the alignment between the clean component's cumulative gradient $G(\mathcal{I})$ and d_{act} without conditioning on the gradient of a specific camouflage component.

At step $j \in \{1, \ldots, k_{\text{cln}}\}$, let $\mathcal{A}_{j-1}$ contain the indices of unselected candidates in $\mathcal{J}$ whose class count in $\mathcal{I}_{j-1}$ is below q. The next candidate maximizes the alignment of the updated cumulative gradient:

$$\begin{aligned} i_j &\in \arg\max_{i \in \mathcal{A}_{j-1}} \cos(G_{j-1} + u_i, d_{\text{act}}), \\ \mathcal{I}_j &= \mathcal{I}_{j-1} \cup \{i_j\},\ G_j = G_{j-1} + u_{i_j}. \end{aligned} \tag{12}$$

After k_{cln} steps, the greedy output defines the selected clean component $\mathcal{D}^{\star}_{\text{cln}} = \{z_i \mid i \in \mathcal{I}_{k_{\text{cln}}}\}$. The final forget set is $\mathcal{D}_f = \mathcal{D}_{\text{cam}} \cup \mathcal{D}^{\star}_{\text{cln}}$. Algorithm 1 in the appendix summarizes the complete selection and request construction procedure.

5 Experiments

5.1 Experimental Setup

Datasets and architectures. CoForget is evaluated on CIFAR-10, CIFAR-100 (Krizhevsky, 2009), and Tiny-ImageNet (Le & Yang, 2015). The primary evaluation employs PreActResNet-18 (He et al., 2016b) as the victim architecture, whereas cross-architecture evaluations additionally incorporate DenseNet-121 (Huang et al., 2017) and ResNet-34 (He et al., 2016a).

Attack and unlearning protocols. The attacker contributes 10% of the training set, with a target-class poisoning ratio of 0.5% across all three datasets. The ℓ_∞ trigger budget is set to $8/255$ for CIFAR-10 and $12/255$ for CIFAR-100 and Tiny-ImageNet. Each unlearning request comprises 0.05% camouflage samples and 0.5% selected clean samples. We consider three representative unlearning algorithms: NegGrad+ (Kurmanji et al., 2023), SAM (Tang & Khanna, 2026), and SCRUB (Kurmanji et al., 2023). For each attacker seed, the identical trigger and preconstructed request are evaluated across all unlearning algorithms. Appendix A provides additional experimental details.

5.2 Attack Effectiveness

Table 1: Effectiveness of CoForget across datasets and unlearning algorithms.

Dataset	Metric	Before unlearning	After unlearning		
			NegGrad+	SAM	SCRUB
CIFAR-10	BA	93.750 ± 0.017	92.487 ± 0.310	92.713 ± 0.569	92.230 ± 0.515
	ASR	12.228 ± 3.325	98.977 ± 1.746	99.997 ± 0.006	99.903 ± 0.159
CIFAR-100	BA	70.610 ± 0.131	68.180 ± 0.838	67.467 ± 0.971	67.620 ± 0.359
	ASR	19.335 ± 3.642	99.227 ± 0.703	98.287 ± 1.354	99.856 ± 0.162
Tiny-ImageNet	BA	56.823 ± 0.206	53.467 ± 0.641	52.863 ± 1.192	53.233 ± 1.009
	ASR	7.149 ± 2.417	97.873 ± 1.737	97.717 ± 0.639	99.130 ± 0.148

Transfer across unlearning algorithms. Table 1 reports the mean and standard deviation over three independent attacker seeds, showing that post-unlearning ASR increases substantially across all algorithms while the BA remains high. The evaluated algorithms span diverse paradigms, including forget losses with original labels, sharpness-aware optimization, and teacher–student distillation. The consistent activation across these distinct methods indicates that the attack is not confined to a specific unlearning objective or update rule, thereby demonstrating its transferability across the evaluated approximate unlearning algorithms.

Transfer across victim architectures. To evaluate the transferability across victim architectures, we construct requests using a PreActResNet-18 shadow model and apply them to PreActResNet-18, DenseNet-121, and ResNet-34 victim models across all datasets. As illustrated in Figure 2, the post-unlearning ASR increases substantially across all settings. The consistent activation across these distinct model families demonstrates the robust cross-architecture transferability of the generated requests. Supplementary numerical results are provided in Appendix Table 11.

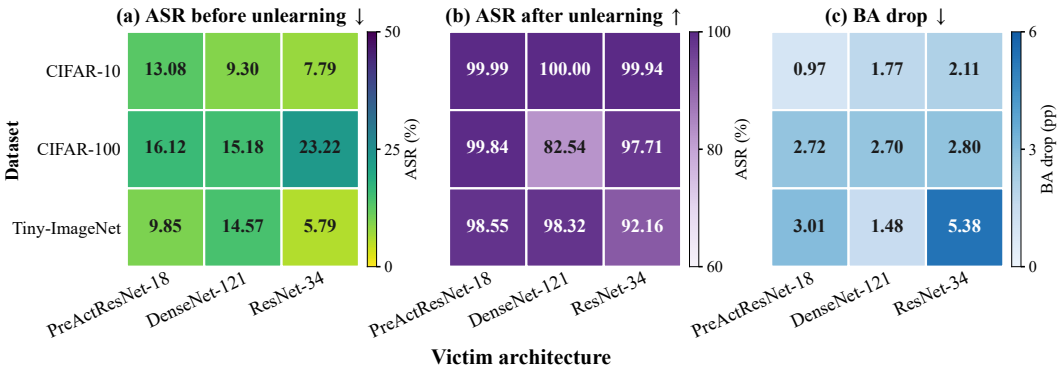


Figure 2: Effectiveness of CoForget across architectures.

Visual appearance. CoForget operates as a clean-label backdoor: both target poisons and camouflage samples retain their original labels. Figure 3 compares clean images with their triggered counterparts across three datasets, allowing the visual effect of the trigger to be directly compared. The triggered images preserve recognizable semantic content while exhibiting only subtle visual differences.

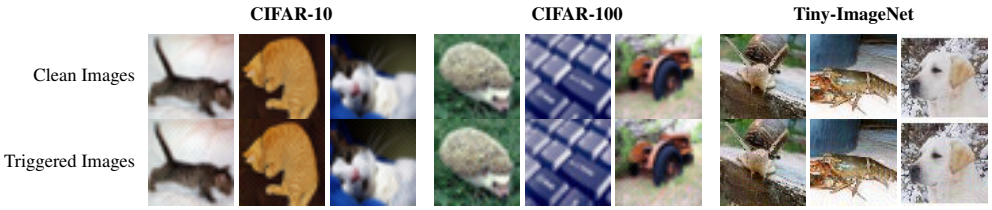


Figure 3: Visual appearance of clean images (top) and their triggered counterparts (bottom).

5.3 Comparison with Prior Unlearning-Activated Backdoor Attacks

CoForget is evaluated alongside three representative unlearning-activated backdoor attacks: UBA-Inf (Huang et al., 2024), ReVeil (Alam et al., 2025), and BAU (Zhang et al., 2023). Whereas these prior attacks construct requests composed solely of camouflage samples, CoForget couples camouflage samples with selected clean samples. As reported in Table 2, the primary evaluation matches the total request budget across all attacks under NegGrad+. Results are reported for a fixed attacker seed.

Table 2: Comparison with prior unlearning-activated backdoor attacks.

Method	CIFAR-10			CIFAR-100			Tiny-ImageNet		
	ASR before	ASR after	ΔBA	ASR before	ASR after	ΔBA	ASR before	ASR after	ΔBA
CoForget	13.080	99.990	0.970	16.120	99.840	2.720	9.849	98.550	3.010
UBA-Inf	6.800	100.000	2.410	11.393	100.000	3.580	14.410	44.710	2.520
ReVeil	9.710	100.000	2.950	10.060	100.000	3.560	16.380	100.000	3.380
BAU	10.411	100.000	2.520	9.535	100.000	3.900	16.382	100.000	4.750

Across all datasets, CoForget achieves a post-unlearning ASR exceeding 98.5% while using 1/11 as many camouflage samples as the prior attacks. On CIFAR-10 and CIFAR-100, it incurs smaller BA degradation, though the prior attacks maintain lower pre-unlearning ASR. On Tiny-ImageNet, CoForget exhibits the lowest pre-unlearning ASR and incurs less BA degradation than ReVeil and BAU, alongside a slightly lower post-unlearning ASR. While UBA-Inf yields a smaller BA reduction on this dataset, its post-unlearning ASR is limited to 44.71%. Extended baseline comparisons are provided in Appendix C.2.

5.4 Ablation Studies

The following ablations examine trigger construction, the contribution of selected clean samples, and the choice of selection strategy.

Table 3: Component ablation of trigger construction.

Dataset	Variant	Before unlearning		After unlearning	
		BA (%)	ASR (%)	BA (%)	ASR (%)
CIFAR-10	**CoForget**	93.750 ± 0.017	12.228 ± 3.325	92.487 ± 0.310	98.977 ± 1.746
	w/o inward initialization	93.757 ± 0.116	14.417 ± 8.655	91.943 ± 0.973	70.800 ± 50.481
	w/o calibration	93.797 ± 0.208	1.300 ± 0.342	90.677 ± 1.402	9.960 ± 2.505
CIFAR-100	**CoForget**	70.610 ± 0.131	19.335 ± 3.642	68.180 ± 0.838	99.227 ± 0.703
	w/o inward initialization	71.047 ± 0.266	18.210 ± 21.890	67.113 ± 1.276	42.303 ± 51.150
	w/o calibration	70.723 ± 0.343	2.550 ± 0.704	66.213 ± 0.759	14.633 ± 19.323
Tiny-ImageNet	**CoForget**	56.823 ± 0.206	7.149 ± 2.417	53.467 ± 0.641	97.873 ± 1.737
	w/o inward initialization	56.670 ± 0.236	7.953 ± 7.799	51.400 ± 0.297	51.640 ± 48.296
	w/o calibration	56.817 ± 0.485	11.447 ± 1.675	53.240 ± 0.121	95.517 ± 1.357

Trigger construction. We ablate inward initialization and logit gap calibration under NegGrad+ (Table 3). Without calibration, pre-unlearning ASR decreases on both CIFAR datasets, but post-unlearning activation drops sharply. On Tiny-ImageNet, the uncalibrated variant retains high post-unlearning ASR but exhibits weaker dormancy before unlearning. Removing inward initialization substantially increases ASR variability both before and after unlearning, while also lowering mean post-unlearning ASR. These findings indicate that calibration effectively balances dormancy with activation, while inward initialization improves their consistency across seeds.

Table 4: Contribution of selected clean samples across unlearning algorithms on CIFAR-10.

Algorithm	Request	BA	ASR
NegGrad+	**+ Selected clean (CoForget)**	92.487 ± 0.310	98.977 ± 1.746
	+ Random clean	91.389 ± 0.942	73.813 ± 26.012
	Camouflage only	86.650 ± 5.581	99.977 ± 0.032
SAM	**+ Selected clean (CoForget)**	92.713 ± 0.569	99.997 ± 0.006
	+ Random clean	90.652 ± 2.455	94.293 ± 9.878
	Camouflage only	92.283 ± 1.051	78.180 ± 37.793
SCRUB	**+ Selected clean (CoForget)**	92.230 ± 0.515	99.903 ± 0.159
	+ Random clean	91.163 ± 1.175	91.418 ± 14.457
	Camouflage only	88.880 ± 3.507	99.980 ± 0.020

Contribution of selected clean samples. Table 4 compares selected clean samples with camouflage-only and random clean baselines. Across all unlearning algorithms, selected clean samples achieve higher mean BA and ASR with lower variability than random clean samples. Compared with camouflage-only requests, they improve mean BA while maintaining high ASR under NegGrad+ and SCRUB; under SAM, they also improve ASR. Overall, selected clean samples yield a more favorable balance between activation and benign utility. Additional results across datasets are provided in Appendix D.2.

Table 5: Comparison of clean sample selection strategies.

Selector	CIFAR-10		CIFAR-100		Tiny-ImageNet	
	BA	ASR	BA	ASR	BA	ASR
Cumulative greedy	92.487 ± 0.310	98.977 ± 1.746	68.180 ± 0.838	99.227 ± 0.703	53.467 ± 0.641	97.873 ± 1.737
Individual top-k	90.750 ± 1.132	72.490 ± 42.096	64.120 ± 5.533	96.303 ± 2.281	52.913 ± 1.239	88.977 ± 16.194
Continuous relaxation	91.817 ± 0.934	70.930 ± 47.581	66.387 ± 2.936	87.443 ± 17.122	52.343 ± 2.096	94.763 ± 2.705

Selection strategies. To assess the clean sample selection strategies, we compare cumulative greedy selection with individual top-k selection and continuous relaxation across the three datasets under NegGrad+. As shown in Table 5, cumulative greedy selection achieves higher mean BA and ASR with smaller standard deviations. These findings support cumulative greedy selection as the default strategy.

5.5 Defense Evaluation

Potential defenses against CoForget are evaluated across three dimensions: data detection, request screening, and model repair. Supplementary detection analyses and model repair results are provided in Appendix F.

Table 6: Detection of target poisons and camouflage samples across datasets.

Dataset	Detector	Target poison		Camouflage	
		TPR (%) @ 1% FPR	TPR (%) @ 5% FPR	TPR (%) @ 1% FPR	TPR (%) @ 5% FPR
CIFAR-10	PSBD	0.40	3.60	0.00	0.00
	Spectral Signatures	0.00	0.00	0.00	0.00
CIFAR-100	PSBD	0.00	0.00	8.00	8.00
	Spectral Signatures	0.00	1.60	0.00	0.00
Tiny-ImageNet	PSBD	0.00	0.60	0.00	0.00
	Spectral Signatures	0.20	1.40	0.00	2.00

Data detection. Data detection is evaluated using PSBD (Li et al., 2025) and Spectral Signatures (Tran et al., 2018). Decision thresholds are calibrated on clean data at nominal false positive rates (FPRs) of 1% and 5%. As shown in Table 6, neither detector achieves a true positive rate (TPR) above 3.6% for target poison samples or 8.0% for camouflage samples across three datasets. Thus, both detectors leave the vast majority of these samples unflagged at the evaluated operating points.

Table 7: BA and ASR under MU request screening across datasets. Remaining counts are reported as camouflage + selected clean samples after screening.

Dataset	No screening		MU: 10% removal			MU: 20% removal		
	BA	ASR	Remaining	BA	ASR	Remaining	BA	ASR
CIFAR-10	92.790	99.990	$22 + 225$	92.420	98.630	$21 + 199$	93.020	99.980
CIFAR-100	67.980	99.840	$23 + 224$	64.380	100.000	$21 + 199$	65.060	100.000
Tiny-ImageNet	54.010	98.550	$41 + 454$	52.310	96.800	$34 + 406$	55.470	97.360

Request screening. To test whether sample-level filtering prevents activation, we rank request samples by predictive Gini impurity, following the Model Uncertainty (MU) criterion from BAU (Zhang et al., 2023). The highest-scoring 10% or 20% are excluded from the request, and NegGrad+ is applied to unlearn the remaining samples. Table 7 shows that ASR remains high across all datasets at both removal rates. Increasing removal to 20% provides no additional suppression.

6 Conclusion

We introduced CoForget, a request-composition attack that combines a small camouflage component with selected clean samples. It calibrates a universal trigger and greedily selects the clean subset through cumulative gradient alignment. Experiments across three datasets and three approximate unlearning algorithms show dormancy before unlearning and strong activation afterward, with limited degradation in BA. Ablations further show that selected clean samples improve mean BA over camouflage-only requests while maintaining high ASR. Together, these results show that unmodified clean samples can contribute to backdoor activation when strategically combined with camouflage samples. This motivates defenses that complement sample screening with an assessment of request composition and the resulting model update.

AI USE STATEMENT

In this work, generative AI tools, including OpenAI ChatGPT, were used to assist with language editing and presentation, including translation, improving readability, and revising wording. These tools were not used to generate datasets or produce the reported numerical results. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final manuscript.

ETHICS STATEMENT

This work investigates a dual-use security risk in approximate machine unlearning. The proposed request-composition procedure could potentially be misused to activate dormant backdoors through apparently legitimate unlearning requests. Our purpose is to characterize this failure mode under an explicit threat model and to draw attention to the security implications of request-level aggregate effects. All experiments were conducted offline using publicly available image classification benchmarks and locally trained research models; no deployed service was targeted, and no private user data or human participants were involved. We evaluate representative screening and repair mechanisms and encourage service providers to complement sample-level inspection with request-provenance checks, contribution controls, aggregate-update auditing, and post-unlearning validation. The results and any accompanying artifacts are intended solely for authorized research, reproducibility, and the development of safer machine unlearning systems.

REPRODUCIBILITY STATEMENT

Section 3 specifies the problem setting and threat model, and Section 4 presents the trigger calibration and clean selection objectives. Appendix A describes the public datasets, preprocessing, victim training and attack configurations, request construction, unlearning algorithms, evaluation metrics, and the computational environment. Algorithm 1 provides the clean selection procedure. Section 5.1 specifies the common evaluation protocol. Appendices C–F provide additional numerical results, baseline comparison protocols, ablations, sensitivity analyses, and defense evaluations.

REFERENCES

Manaar Alam, Hithem Lamri, and Michail Maniatakos. ReVeil: Unconstrained concealed backdoor attack on deep neural networks using machine unlearning. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pp. 1–7, 2025.

Marco Arazzi, Antonino Nocera, and Vinod P. When forgetting triggers backdoors: A clean unlearning attack. *arXiv preprint arXiv:2506.12522*, 2025.

Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in CNNs by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing*, pp. 101–105, 2019.

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.

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

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Jimmy Di, Jack Douglas, Jayadev Acharya, Gautam Kamath, and Ayush Sekhari. Hidden poison: Machine unlearning enables camouflaged poisoning attacks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44424–44450, 2023.

Ali Ebrahimpour-Boroojeny, Hari Sundaram, and Varun Chandrasekaran. Not all wrong is bad: Using adversarial examples for unlearning. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 14950–14971, 2025.

Antonio Ginart, Melody 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, pp. 3513–3526, 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.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

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.

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, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016*, pp. 630–645, 2016b.

Hongsheng Hu, Shuo Wang, Jiamin Chang, Haonan Zhong, Ruoxi Sun, Shuang Hao, Haojin Zhu, and Minhui Xue. A duty to forget, a right to be assured? Exposing vulnerabilities in machine unlearning services. In *Proceedings of the 31st Annual Network and Distributed System Security Symposium (NDSS)*, 2024.

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.

Yangsibo Huang, Daogao Liu, Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Milad Nasr, Amer Sinha, and Chiyuan Zhang. Unlearn and burn: Adversarial machine unlearning requests destroy model accuracy. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zirui Huang, Yunlong Mao, and Sheng Zhong. UBA-Inf: Unlearning activated backdoor attack with Influence-Driven camouflage. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 4211–4228, 2024.

Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, pp. 2008–2016, 2021.

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, pp. 1957–1987, 2023.

Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. Stanford CS231N Course Project Report, 2015. URL `https://cs231n.stanford.edu/reports/2015/pdfs/yle_project.pdf`.

Wei Li, Pin-Yu Chen, Sijia Liu, and Ren Wang. PSBD: Prediction shift uncertainty unlocks backdoor detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10255–10264, 2025.

Zihao Liu, Tianhao Wang, Mengdi Huai, and Chenglin Miao. Backdoor attacks via machine unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14115–14123, 2024.

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*, pp. 931–962, 2021.

Jie Peng, Hongwei Yang, Jing Zhao, Hengji Dong, Hui He, Weizhe Zhang, and Haoyu He. Circumventing backdoor space via weight symmetry. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 48886–48922, 2025.

Wei Qian, Chenxu Zhao, Wei Le, Meiyi Ma, and Mengdi Huai. Towards understanding and enhancing robustness of deep learning models against malicious unlearning attacks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1932–1942, 2023.

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, pp. 18075–18086, 2021.

Haoran Tang and Rajiv Khanna. Sharpness-aware machine unlearning. In *International Conference on Learning Representations*, pp. 79941–79985, 2026.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Shaokui Wei, Jiayin Liu, and Hongyuan Zha. Backdoor mitigation by distance-driven detoxification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4465–4474, 2025.

Yi Zeng, Minzhou Pan, Hoang Anh Just, Lingjuan Lyu, Meikang Qiu, and Ruoxi Jia. Narcissus: A practical clean-label backdoor attack with limited information. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 771–785, 2023.

Peixin Zhang, Jun Sun, Mingtian Tan, and Xinyu Wang. Exploiting machine unlearning for backdoor attacks in deep learning system. *arXiv preprint arXiv:2310.10659*, 2023.

Dongdong Zhao, Can Li, Xiang Yao, Fan He, Qihang Ge, and Baogang Song. Benign on label, malicious by design: Clean-label dormant-to-activated backdoor via machine unlearning with removable camouflage. *arXiv preprint arXiv:2607.27936*, 2026.

APPENDIX CONTENTS

A Experimental Setup and Implementation Details **14**
A.1 Datasets, Models, and Victim Training 14
A.2 Attack Configuration 14
A.3 Selector Implementation 14
A.4 Approximate Unlearning Algorithms 16
A.5 Metrics and Computational Environment 16

B Local Rationale for Cumulative Clean Selection **16**

C Extended Attack Evaluation **17**
C.1 Transfer Across Architectures 17
C.2 Extended Baseline Comparisons 17

D Ablation Studies **18**
D.1 Alternative Trigger Designs 18
D.2 Contribution of the Clean Component 19

E Sensitivity Analyses **19**
E.1 Request Composition 19
E.2 Total Unlearning Request Budget 20
E.3 Trigger Calibration 20
E.4 Attacker Data Availability 21

F Additional Defense Evaluation **21**
F.1 Training Data Detection 21
F.2 Model Repair 22

A Experimental Setup and Implementation Details

A.1 Datasets, Models, and Victim Training

We adopt the standard splits of CIFAR-10, CIFAR-100, and Tiny-ImageNet. CIFAR-10 and CIFAR-100 each comprise 50,000 training images and 10,000 test images across 10 and 100 classes, respectively. Tiny-ImageNet contains 100,000 training images and 10,000 validation images across 200 classes. For evaluation, we use the test sets for the two CIFAR datasets and the validation set for Tiny-ImageNet. All image channels are normalized using a mean of 0.5 and a standard deviation of 0.5.

We use PreActResNet-18 as the default victim architecture. The victim models are trained using SGD with momentum 0.9, weight decay 5×10^{-4}, and an initial learning rate of 0.01. Dataset-specific training epochs, batch sizes, and learning-rate schedules are given in Table 8. For cross-architecture evaluation, we use ResNet-34 and DenseNet-121 as alternative victim architectures while retaining the same trigger and unlearning request constructed using the PreActResNet-18 shadow model.

Table 8: Victim training configurations.

Dataset	Architecture	Epochs	Batch	Optimizer	Scheduler	Target
CIFAR-10	PreActResNet-18	110	128	SGD	Cosine annealing	6
CIFAR-100	PreActResNet-18	100	128	SGD	Cosine annealing	9
Tiny-ImageNet	PreActResNet-18	120	64	SGD	Reduce on plateau	6

A.2 Attack Configuration

For each dataset, the attacker samples 10% of the unmodified training split to form the attacker data $\mathcal{D}_A$. Sampling on CIFAR-10 is conducted uniformly without replacement. For CIFAR-100 and Tiny-ImageNet, target class samples are reserved first within this budget, and the remaining slots are filled by uniformly sampling non-target samples without replacement. All sampling procedures are controlled by the designated attacker seed. The source model is trained on Tiny-ImageNet data for the two CIFAR datasets, and on CIFAR-100 data for Tiny-ImageNet. We adapt the source model into a K-class shadow model by replacing its classification head and fine-tuning on the attacker data $\mathcal{D}_A$. This adaptation precedes trigger calibration and uses the dataset-specific epoch counts in Table 9.

We use Adam with a learning rate of 5×10^{-4} for both trigger initialization and calibration, projecting the trigger onto the specified ℓ_∞ ball after each update. The calibration-loss and anchor-loss weights are set to $\lambda_{\text{cal}} = 1.6$ and $\lambda_{\text{anc}} = 0.3$, respectively. During victim training, poisoned samples drawn from the target class constitute 0.5% of each training set.

Table 9: Attack configuration for each dataset.

Dataset	Source data	ϵ	Adaptation epochs	$[m_{\text{low}}, m_{\text{high}}]$
CIFAR-10	Tiny-ImageNet	8/255	5	[0.2, 1.0]
CIFAR-100	Tiny-ImageNet	12/255	5	[1.0, 1.5]
Tiny-ImageNet	CIFAR-100	12/255	3	[0.2, 1.0]

A.3 Selector Implementation

The clean candidate pool $\mathcal{C}$ is sampled uniformly without replacement from the non-target attacker data, excluding samples assigned to the camouflage component. A fixed set $\mathcal{D}_{\text{nt}}$ of 4,000 samples from the same non-target attacker data pool is used to estimate d_{act}; overlap with $\mathcal{C}$ is allowed. We estimate d_{act} on $\mathcal{D}_{\text{nt}}$ using the calibrated trigger and target label, while candidate gradients are

computed on clean samples with their original labels. All gradients are taken with respect to the final two residual stages and classification head of the shadow PreActResNet-18. A coarse filtering step retains the m candidates with the largest $\cos(u_i, d_{\text{act}})$, after which their Gram matrix and directional inner products are precomputed for greedy selection.

Initialized with $\mathcal{I}_0 = \emptyset$ and $G_0 = 0$, the greedy procedure iteratively selects k_{cln} samples by maximizing the cumulative cosine alignment with d_{act} at each step, subject to the per-class cap q. The cached inner products enable incremental scoring and updates with $O(m)$ operations per iteration. The coarse pool sizes and per-class caps are listed in Table 10, and the complete procedure is summarized in Algorithm 1.

Algorithm 1 Clean sample selection and heterogeneous request construction

Require: Shadow model $f_{\text{sh}}(\cdot; \theta_{\text{sh}})$ and selected parameters ϕ_{sh}
Require: Calibrated trigger $\delta^\star$, target label t, and non-target set $\mathcal{D}_{\text{nt}}$
Require: Clean candidate pool $\mathcal{C} = \{z_i = (x_i, y_i)\}_{i=1}^{M}$ and fixed camouflage set $\mathcal{D}_{\text{cam}}$
Require: Coarse pool size m, clean selection budget k_{cln}, and class cap q
Ensure: Selected clean component $\mathcal{D}^\star_{\text{cln}}$ and final forget set $\mathcal{D}_f$

$\mathcal{L}_{\text{bd}} \leftarrow \dfrac{1}{|\mathcal{D}_{\text{nt}}|} \displaystyle\sum_{(x,y)\in\mathcal{D}_{\text{nt}}} \ell(f_{\text{sh}}(T_{\delta^\star}(x); \theta_{\text{sh}}), t)$
$d_{\text{act}} \leftarrow -\nabla_{\phi_{\text{sh}}} \mathcal{L}_{\text{bd}}$
for $i = 1, \ldots, M$ **do**
    $u_i \leftarrow \nabla_{\phi_{\text{sh}}} \ell(f_{\text{sh}}(x_i; \theta_{\text{sh}}), y_i)$
    $s_i \leftarrow \cos(u_i, d_{\text{act}})$
end for
$\mathcal{J} \leftarrow$ indices of the m largest values in $\{s_i\}_{i=1}^{M}$
Precompute $\langle u_i, d_{\text{act}} \rangle$ and $\langle u_i, u_{i'} \rangle$ for all $i, i' \in \mathcal{J}$
$\mathcal{I}_0 \leftarrow \emptyset$ and $G_0 \leftarrow 0$
for $j = 1, \ldots, k_{\text{cln}}$ **do**
    $\mathcal{A}_{j-1} \leftarrow \{i \in \mathcal{J} \setminus \mathcal{I}_{j-1} \mid |\{k \in \mathcal{I}_{j-1} \mid y_k = y_i\}| < q\}$
    $i_j \leftarrow \arg\max_{i\in\mathcal{A}_{j-1}} \cos(G_{j-1} + u_i, d_{\text{act}})$
    $\mathcal{I}_j \leftarrow \mathcal{I}_{j-1} \cup \{i_j\}$
    $G_j \leftarrow G_{j-1} + u_{i_j}$
end for
$\mathcal{D}^\star_{\text{cln}} \leftarrow \{z_i \mid i \in \mathcal{I}_{k_{\text{cln}}}\}$
$\mathcal{D}_f \leftarrow \mathcal{D}_{\text{cam}} \cup \mathcal{D}^\star_{\text{cln}}$
return $\mathcal{D}^\star_{\text{cln}}, \mathcal{D}_f$

Table 10: Configuration of the cumulative greedy selector for each dataset.

Dataset	Coarse pool m	Class cap q
CIFAR-10	3000	60
CIFAR-100	3000	60
Tiny-ImageNet	3000	20

Alternative selection strategies. For the comparison of selection strategies, we implement individual top-k selection and continuous relaxation as alternatives to cumulative greedy selection. Within each attacker seed, all three strategies share the shadow checkpoint, coarse-filtered candidate pool $\mathcal{J}$, parameter subspace, activation direction, and clean selection budget.

(i) Individual top-k. The k_{cln} candidates with the highest $\cos(u_i, d_{\text{act}})$ are selected using fixed individual scores, without accounting for the selected subset.

(ii) Continuous relaxation. Candidate logits are initialized as $a_i^{(0)} = \cos(u_i, d_{\text{act}})$. With weights parameterized as $w_i = k_{\text{cln}} \operatorname{softmax}(a/\tau)_i$, the optimization formulation minimizes $-\cos(\sum_{i\in\mathcal{J}} w_i u_i, d_{\text{act}})$. These weights remain nonnegative and sum to k_{cln}. The logits are op-

timized via gradient descent for 200 steps with a learning rate of 0.1, accompanied by exponential temperature annealing from 1.0 to 0.05. At the final temperature, the optimized logits are converted into weights, and the k_{cln} candidates with the largest weights form the discrete clean subset.

A.4 Approximate Unlearning Algorithms

We evaluate three approximate unlearning algorithms representing distinct paradigms for incorporating the forget signal: loss reversal, sharpness-aware optimization, and teacher–student distillation. The respective update principles of these methods are outlined below.

- **NegGrad+** (Kurmanji et al., 2023) combines gradient descent on the cross-entropy loss over the retain set with gradient ascent on the forget set.
- **SAM** (Tang & Khanna, 2026) applies sharpness-aware optimization to an objective combining retain cross-entropy with negative forget cross-entropy. The gradient at the current parameters determines a normalized local perturbation, after which the gradient is recomputed at the perturbed parameters and used to update the original parameters.
- **SCRUB** (Kurmanji et al., 2023) freezes the original model prior to unlearning as a teacher and updates a student model via opposing distillation objectives. Specifically, it maximizes the divergence between the teacher and student models on forget samples, while minimizing distillation and supervised losses on retain samples.

A.5 Metrics and Computational Environment

Benign accuracy is the top-1 accuracy on the clean evaluation set. Attack success rate is the fraction of triggered non-target evaluation samples classified as the target class. Both metrics are evaluated before and after unlearning. We define $\Delta\mathrm{BA}$ as BA before unlearning minus BA after unlearning, measured in percentage points.

Experiments were implemented in Python 3.8.20 with PyTorch 1.11.0+cu113, using the PyTorch CUDA runtime 11.3 and cuDNN 8.2.0. Computations were performed on a server equipped with four NVIDIA GeForce RTX 4090 GPUs, each with 24 GB of memory, using NVIDIA driver 595.71.05.

B Local Rationale for Cumulative Clean Selection

This section examines the clean component proxy in Section 4.2 around a fixed shadow-parameter reference point. Let ϕ_0 denote the reference value of the parameter subvector ϕ_{sh}, and let $F(\phi)$ denote $\mathcal{L}_{\mathrm{bd}}$ in Equation 8 viewed as a function of this subvector. The remaining parameters, model buffers, calibrated trigger, and estimation set are held fixed. All candidate gradients u_i in Equation 9 are evaluated at ϕ_0, with $G(\mathcal{I}) = \sum_{i\in\mathcal{I}} u_i$.

Directional interpretation. Assume that F is differentiable at ϕ_0 and $d_{\mathrm{act}} = -\nabla F(\phi_0) \neq 0$. For a clean subset with $G(\mathcal{I}) \neq 0$, define the unit direction $v_{\mathcal{I}} = G(\mathcal{I})/\|G(\mathcal{I})\|_2$. By the chain rule,

$$-\frac{\mathrm{d}}{\mathrm{d}\rho}F(\phi_0 + \rho v_{\mathcal{I}})\bigg|_{\rho=0} = \frac{\langle d_{\mathrm{act}}, G(\mathcal{I})\rangle}{\|G(\mathcal{I})\|_2} = \|d_{\mathrm{act}}\|_2 \cos(G(\mathcal{I}), d_{\mathrm{act}})\,. \tag{13}$$

Thus, positive cosine similarity gives a local descent direction for the target-label loss under a sufficiently small positive displacement. The inner product in the unnormalized proxy step reflects both alignment and the magnitude of the cumulative gradient. Cosine similarity normalizes the cumulative gradient magnitude, emphasizing directional alignment.

Displacement interpretation. For a subset with $\cos(G(\mathcal{I}), d_{\mathrm{act}}) > 0$, consider a prescribed first-order loss reduction $\gamma > 0$. In the linearized model at ϕ_0, the corresponding displacement magnitude along $v_{\mathcal{I}}$ is

$$\rho^{\mathrm{lin}}_{\mathcal{I}} = \frac{\gamma}{\|d_{\mathrm{act}}\|_2 \cos(G(\mathcal{I}), d_{\mathrm{act}})}. \tag{14}$$

At the same reference point, a larger cosine therefore corresponds to a smaller displacement for the same first-order loss reduction.

Dependence on the selected subset. For a cumulative gradient G, adding a feasible candidate u_i changes both the projection onto d_{act} and the norm of the cumulative gradient. Whenever $G + u_i \neq 0$,

$$\cos(G + u_i, d_{\text{act}}) = \frac{\langle G, d_{\text{act}} \rangle + \langle u_i, d_{\text{act}} \rangle}{\|d_{\text{act}}\|_2 \sqrt{\|G\|_2^2 + 2\langle G, u_i \rangle + \|u_i\|_2^2}}. \tag{15}$$

The interaction term $\langle G, u_i \rangle$ captures the dependence between each candidate and the current cumulative gradient. As a result, the candidate score depends on the subset already selected, even though the candidate gradients themselves remain fixed. Individual cosine rankings therefore do not generally determine the ranking of candidate additions under the cumulative objective. CoForget accounts for this dependence by scoring each feasible addition using the updated cumulative gradient, starting from the empty clean subset with $G_0 = 0$.

These identities provide a local rationale for the cosine objective and its cumulative evaluation on the shadow model. Section 5.4 evaluates the resulting selection strategies through backdoor activation and benign accuracy under approximate unlearning.

C Extended Attack Evaluation

C.1 Transfer Across Architectures

Table 11: Cross-architecture transfer results across datasets.

Dataset	Victim architecture	Before unlearning		After unlearning	
		BA	ASR	BA	ASR
CIFAR-10	PreActResNet-18	93.760	13.080	92.790	99.990
	DenseNet-121	93.700	9.300	91.930	100.000
	ResNet-34	93.950	7.788	91.840	99.940
CIFAR-100	PreActResNet-18	70.700	16.120	67.980	99.840
	DenseNet-121	70.200	15.180	67.500	82.540
	ResNet-34	71.750	23.220	68.950	97.710
Tiny-ImageNet	PreActResNet-18	57.020	9.849	54.010	98.550
	DenseNet-121	61.550	14.570	60.070	98.320
	ResNet-34	59.450	5.789	54.070	92.160

The numerical results underlying Figure 2 are provided in Table 11. Across the six cross-architecture settings in Table 11, post-unlearning ASR exceeds 92% in five settings and reaches 82.54% on CIFAR-100 with DenseNet-121. Meanwhile, BA decreases by less than 3 percentage points in five settings. These results show that the trigger and request constructed on the PreActResNet-18 shadow model transfer effectively to both alternative victim architectures while preserving benign utility.

C.2 Extended Baseline Comparisons

Baseline protocols. UBA-Inf, ReVeil, and BAU use BadNets triggers, which were evaluated in all three studies. Each baseline retains its original camouflage construction procedure and perturbation settings; only the camouflage count is adjusted according to the two budget controls below.

CoForget uses camouflage and selected clean samples amounting to 0.05% and 0.5% of the training set, respectively. Baseline requests contain only camouflage samples: 0.55% of the training set to match the total request size, or 0.05% to match the camouflage count. All results use a fixed attacker seed and are reported in Tables 12–14.

When matching the total request size, CoForget maintains post-unlearning ASR above 96% across all datasets. On both CIFAR datasets, it incurs the smallest BA degradation under NegGrad+. On Tiny-ImageNet, CoForget maintains activation under SAM, where all three baselines exhibit low post-unlearning ASR. These results demonstrate that CoForget achieves effective activation with fewer camouflage samples under the same total request budget.

When matching the camouflage count, CoForget combines lower pre-unlearning ASR with high post-unlearning ASR. The baselines exhibit substantial backdoor activity before unlearning, and

some CIFAR-100 settings also show weak post-unlearning activation. These results support CoForget's ability to maintain dormancy and enable subsequent activation with a small camouflage component.

Table 12: Comparison with prior attacks under two budget controls on CIFAR-10.

Protocol	Method	Before unlearning		NegGrad+		SAM		SCRUB	
		BA	ASR	BA	ASR	BA	ASR	BA	ASR
Request size	**CoForget**	93.760	13.080	92.790	99.990	92.380	99.990	92.220	100.000
	UBA-Inf	93.450	6.800	91.040	100.000	92.010	100.000	91.670	100.000
	ReVeil	93.500	9.710	90.550	100.000	92.300	100.000	91.330	100.000
	BAU	93.710	10.411	91.190	100.000	92.480	100.000	92.040	100.000
Camouflage count	**CoForget**	93.760	13.080	92.790	99.990	92.380	99.990	92.220	100.000
	UBA-Inf	93.580	49.922	92.310	100.000	92.010	100.000	92.130	100.000
	ReVeil	93.580	54.820	91.350	100.000	92.530	100.000	91.880	100.000
	BAU	93.540	53.160	92.070	100.000	91.630	100.000	92.820	100.000

Table 13: Comparison with prior attacks under two budget controls on CIFAR-100.

Protocol	Method	Before unlearning		NegGrad+		SAM		SCRUB	
		BA	ASR	BA	ASR	BA	ASR	BA	ASR
Request size	**CoForget**	70.700	16.120	67.980	99.840	66.440	97.830	67.880	100.000
	UBA-Inf	71.400	11.393	67.820	100.000	66.280	100.000	66.960	100.000
	ReVeil	70.840	10.060	67.280	100.000	67.910	100.000	63.210	100.000
	BAU	70.020	9.535	66.120	100.000	68.320	100.000	67.350	100.000
Camouflage count	**CoForget**	70.700	16.120	67.980	99.840	66.440	97.830	67.880	100.000
	UBA-Inf	70.890	42.050	66.490	0.000	68.530	100.000	68.900	1.670
	ReVeil	70.630	39.840	67.470	100.000	66.910	100.000	68.240	100.000
	BAU	70.930	37.760	68.460	0.000	67.560	0.000	68.110	0.000

Table 14: Comparison with prior attacks under two budget controls on Tiny-ImageNet.

Protocol	Method	Before unlearning		NegGrad+		SAM		SCRUB	
		BA	ASR	BA	ASR	BA	ASR	BA	ASR
Request size	**CoForget**	57.020	9.849	54.010	98.550	54.240	98.310	53.870	98.960
	UBA-Inf	57.050	14.410	54.530	44.710	52.240	12.500	50.620	100.000
	ReVeil	56.350	16.380	52.970	100.000	54.550	5.700	55.990	64.030
	BAU	56.350	16.382	51.600	100.000	52.320	8.160	45.710	100.000
Camouflage count	**CoForget**	57.020	9.849	54.010	98.550	54.240	98.310	53.870	98.960
	UBA-Inf	56.330	49.668	53.830	100.000	55.210	100.000	55.640	99.650
	ReVeil	57.110	56.390	54.180	100.000	53.810	100.000	54.250	100.000
	BAU	55.810	56.391	54.620	100.000	54.510	100.000	54.370	100.000

D Ablation Studies

This section presents two additional ablations: comparisons of alternative trigger designs and an extended evaluation of the clean component across datasets.

D.1 Alternative Trigger Designs

We compare the calibrated trigger with three alternative designs: BadNet (Gu et al., 2017), Blended (Chen et al., 2017), and SIG (Barni et al., 2019). Each candidate trigger is embedded into the heterogeneous unlearning request under the same composition ratios, with all evaluations conducted using NegGrad+.

As shown in Table 15, the calibrated trigger achieves low pre-unlearning ASR and high post-unlearning ASR across all three datasets. BadNet's activation varies substantially across datasets,

Table 15: Comparison with alternative trigger designs across datasets.

Dataset	Trigger	Before unlearning		After unlearning	
		BA	ASR	BA	ASR
CIFAR-10	**CoForget**	93.760	13.080	92.790	99.990
	BadNet	93.800	0.900	91.750	1.110
	Blended	93.760	21.570	90.790	0.110
	SIG	93.790	40.060	89.510	100.000
CIFAR-100	**CoForget**	70.700	16.120	67.980	99.840
	BadNet	70.510	2.340	66.580	99.160
	Blended	70.390	46.750	61.430	88.530
	SIG	70.930	77.970	66.650	99.980
Tiny-ImageNet	**CoForget**	57.020	9.849	54.010	98.550
	BadNet	57.660	4.400	50.630	78.460
	Blended	56.800	42.720	51.910	99.640
	SIG	56.950	57.010	54.540	99.990

whereas Blended and SIG exhibit higher pre-unlearning ASR. Among the evaluated triggers, CoForget achieves low pre-unlearning ASR and high post-unlearning ASR while maintaining relatively small BA degradation.

D.2 Contribution of the Clean Component

For the clean component comparisons in Section 5.4 and this subsection, random and selected clean requests use the same camouflage component and clean sample budget. For the random clean baseline, BA and ASR are first averaged over three independently sampled subsets within each attacker seed; we then report the mean and standard deviation across attacker seeds.

We extend the clean component comparison across all three datasets under NegGrad+. Across all datasets, selected clean samples achieve a more favorable balance between benign accuracy and backdoor activation than random clean samples, while also preserving higher benign accuracy than camouflage-only requests with high post-unlearning ASR.

Table 16: Contribution of selected clean samples across datasets.

Dataset	Request	BA after	ASR after
CIFAR-10	**+ Selected clean (CoForget)**	92.487 ± 0.310	98.977 ± 1.746
	+ Random clean	91.389 ± 0.942	73.813 ± 26.012
	Camouflage only	86.650 ± 5.581	99.977 ± 0.032
CIFAR-100	**+ Selected clean (CoForget)**	68.180 ± 0.838	99.227 ± 0.703
	+ Random clean	67.381 ± 0.709	88.076 ± 14.521
	Camouflage only	67.533 ± 1.420	98.067 ± 1.847
Tiny-ImageNet	**+ Selected clean (CoForget)**	53.467 ± 0.641	97.873 ± 1.737
	+ Random clean	52.584 ± 1.105	82.326 ± 20.127
	Camouflage only	52.567 ± 1.012	97.233 ± 2.849

E Sensitivity Analyses

This section investigates how request composition, total request budget, trigger calibration, and attacker data availability influence backdoor dormancy, activation, and benign utility. All sensitivity analyses use CIFAR-10 and NegGrad+ with a fixed attacker seed. Bold configuration entries indicate the default settings.

E.1 Request Composition

The allocation between the two components of the unlearning request is evaluated under a fixed total budget of $B = 275$. The camouflage count varies from 10 to 100, with the remaining budget allocated to selected clean samples.

Table 17: Sensitivity to request composition with a fixed total budget.

Camouflage	Selected clean	Before unlearning		After unlearning	
		BA	ASR	BA	ASR
10	265	93.740	20.020	91.090	99.990
15	260	93.740	18.520	92.190	100.000
25	**250**	93.730	15.040	92.500	99.980
50	225	93.730	10.860	91.920	99.990
100	175	93.870	13.890	92.410	99.990

Post-unlearning ASR remains high across all tested compositions in Table 17. The default composition achieves the highest BA after unlearning. Increasing the camouflage count to 50 or 100 lowers ASR before unlearning but also reduces BA, illustrating a trade-off between dormancy and benign utility.

E.2 Total Unlearning Request Budget

Building upon the fixed-budget analysis, the total budget is varied while preserving the camouflage-to-clean ratio of 1:10. Request sizes range from 55 to 440 samples, with $B = 275$ as the default.

Table 18: Sensitivity to total unlearning budget at a fixed request ratio.

Scale	B	Request	Before unlearning		After unlearning	
			BA	ASR	BA	ASR
$0.2\times$	55	$5 + 50$	93.680	31.770	92.670	99.980
$0.4\times$	110	$10 + 100$	93.740	20.020	92.890	100.000
$0.6\times$	165	$15 + 150$	93.740	18.520	92.470	100.000
$\mathbf{1.0\times}$	**275**	$\mathbf{25 + 250}$	93.730	15.040	92.500	99.980
$1.6\times$	440	$40 + 400$	93.830	12.440	90.930	100.000

As shown in Table 18, strong activation persists even at the smallest tested request budget. ASR before unlearning decreases as the budget increases, whereas BA after unlearning varies nonmonotonically. The default budget yields lower ASR before unlearning than all smaller requests and higher BA afterward than the largest request. Increasing the budget beyond the default further improves dormancy at the cost of benign utility.

E.3 Trigger Calibration

The role of trigger calibration is examined next. The calibration interval $[m_{\text{low}}, m_{\text{high}}]$ specifies the desired logit gap during trigger construction. We vary this interval across five bands while keeping the unlearning request composition fixed.

Table 19: Sensitivity to the logit gap calibration interval.

Calibration band	Before unlearning		After unlearning	
	BA	ASR	BA	ASR
$[3.0, 4.0]$	93.770	10.460	92.600	94.060
$[2.0, 3.0]$	93.880	13.690	91.040	99.990
$\mathbf{[0.2, 1.0]}$	93.730	15.040	92.500	99.980
$[0.2, 0.6]$	93.910	16.680	92.050	100.000
$[0.1, 0.5]$	93.600	20.470	92.220	100.000

As reported in Table 19, the highest calibration band yields the lowest ASR before unlearning, but at the expense of a reduced post-unlearning ASR of 94.06%. Among the remaining bands, which retain near-complete activation, the default $[0.2, 1.0]$ achieves the highest BA after unlearning, with slightly higher ASR beforehand than $[2.0, 3.0]$. These observations justify the default configuration as an effective trade-off among dormancy, activation, and benign utility.

E.4 Attacker Data Availability

This evaluation investigates whether activation can be maintained with reduced attacker data. Attacker data availability ranges from 4% to 10% of the full training set. As data availability decreases, both the non-target set used to estimate d_{act} and the candidate pool are scaled proportionally. In all cases, the size and composition of the final unlearning request remain fixed.

Table 20: Sensitivity to attacker data availability.

Attacker data	Before unlearning		After unlearning	
	BA	ASR	BA	ASR
4%	93.680	17.790	92.450	100.000
6%	93.960	14.000	91.570	99.960
8%	93.700	19.540	90.780	100.000
10%	93.730	15.040	92.500	99.980

Results in Table 20 show that post-unlearning ASR remains high across all tested levels. Even at 4% attacker data, post-unlearning BA and ASR are comparable to the default setting, while pre-unlearning ASR remains low. No monotonic degradation in benign accuracy is observed as attacker data availability decreases, indicating that CoForget remains effective with substantially less attacker data.

F Additional Defense Evaluation

This section provides supplementary training data detection analyses and evaluates model repair before and after unlearning.

F.1 Training Data Detection

Detection methods and evaluation protocol. Training data detection employs PSBD (Li et al., 2025) and Spectral Signatures (Tran et al., 2018) on the contaminated training set. Specifically, PSBD ranks samples by Prediction Shift Uncertainty, whereas Spectral Signatures computes projection scores derived from class-centered penultimate features. Detection thresholds are calibrated on an independent clean split at nominal FPRs of 1% and 5%. Table 6 reports TPR values at these nominal operating points, whereas Table 21 details the corresponding empirical clean FPRs and AUROC scores.

For AUROC, target poison samples are compared against remaining training samples, whereas camouflage samples are compared against unmodified clean samples. The measured clean FPRs remain close to their nominal targets across datasets. Detection discrimination is also limited: most AUROC values are close to 0.5, and none exceeds 0.6603. Together with the low TPRs in Table 6, these results indicate limited discrimination under both detectors.

Table 21: Supplementary training data detection results for CoForget across datasets.

Dataset	Detector	Actual clean FPR (%)		AUROC (0–1)	
		Nominal 1%	Nominal 5%	Target poison	Camouflage
CIFAR-10	PSBD	1.21	5.40	0.4578	0.5973
	Spectral Signatures	0.76	4.80	0.5015	0.5540
CIFAR-100	PSBD	1.27	4.95	0.3467	0.6603
	Spectral Signatures	1.20	5.31	0.5541	0.5459
Tiny-ImageNet	PSBD	0.66	5.42	0.6040	0.3893
	Spectral Signatures	1.04	5.24	0.4872	0.5000

F.2 Model Repair

Two model repair mechanisms are evaluated: Distance-Driven Detoxification (D3; Wei et al., 2025) applied prior to unlearning, and Two-stage Symmetry Connectivity (TSC; Peng et al., 2025) applied after unlearning.

Table 22: BA and ASR under model repair across datasets.

Dataset	No defense		D3		TSC	
	BA	ASR	BA	ASR	BA	ASR
CIFAR-10	92.790	99.990	92.780	99.990	91.290	90.590
CIFAR-100	67.980	99.840	66.560	99.960	65.900	65.310
Tiny-ImageNet	54.010	98.550	51.720	99.420	51.500	59.210

Table 22 shows that D3 leaves backdoor activation largely intact across all three datasets. Meanwhile, TSC reduces ASR but leaves substantial residual attack success. It also degrades BA relative to undefended models. Consequently, neither defense achieves effective backdoor suppression while preserving benign utility under the evaluated settings.