GUARD: Scalable Gradient-based Unlearning with Adversarial Robustness Defense

Xuechao Lan[1,2,3], **Wenmin Li**[1,2], **Sujuan Qin**[1,2], **Zhengping Jin**[*,1,2], **Fei Gao** [*,1,2]
[1]State Key Laboratory of Networking and Switching Technology
[2] School of Cyberspace Security
[3]National Engineering Research Center of Disaster Backup and Recovery
Beijing University of Posts and Telecommunications, Beijing, 100876, China

Abstract

Machine Unlearning (MU) is an emerging task to remove specified training data from a trained model while preserving its utility, with Gradient-based Unlearning (GU) standing out as a prevalent category due to high scalability. However, standard MU which solely optimizes for clean accuracy, inadvertently damages adversarial robustness on the retain data. Although a few studies extend MU to adversarially trained models, they rely on strong assumptions (e.g., well-conditioned Hessians, smoothness) and costly Hessian-based updates, limiting scalability and efficiency. In this paper, we propose **GUARD**, a scalable and efficient framework that achieves unlearning while preserving retain-data robustness. Having identified the two causes of robustness degradation, we accordingly design *Robustness-Aware Gradient Projection* and *Importance-Guided Parameter Selection*. Both modules rely on a compact *adversarially vulnerable dataset* that we construct with or without retain data, enabling scalability under varying data constraints. Validated by theoretical analysis and experiments across 6 settings, GUARD outperforms 6 baselines, boosting robust accuracy by up to 6.3×, closely matching the Retrain oracle in both unlearning and robustness.

1 Introduction

Machine Unlearning (MU) enables efficient removal of specific data from a trained model [4]. It emerged in response to the General Data Protection Regulation (GDPR) [46], which grants users the right to remove personal data, thus reducing the risk of privacy leakage. MU also serves security purposes by unlearning poisoned data to mitigate poisoning attacks. Ideally, the unlearned model should be indistinguishable from a model retrained from scratch without these data, balancing the erasure of forget data with the preservation of retain utility.

MU is typically classified as exact or approximate based on whether it guarantees strict indistinguishability from the retrain oracle. Due to the high cost of exact MU, research primarily focuses on approximate MU. In particular, Gradient-based Unlearning (GU) [24] is widely adopted for its scalability, which optimizes the unlearning objective using first-order gradients. Based on data usage, GU is categorized as retain-based and retain-free, with the latter being more challenging but practical in real world, where the original retain data is often inaccessible due to privacy or storage constraints.

Since standard MU solely focuses on clean accuracy to optimize unlearning performance, it neglects its detrimental impact on adversarial robustness, as confirmed by [51]. In response, studies [35, 34] achieve unlearning while preserving robustness in adversarially trained models, by using second-order information to account for the indirect influence of inner perturbation variables. However, they

*Corresponding authors: Zhengping Jin <zhpjin@bupt.edu.cn>, Fei Gao <gaof@bupt.edu.cn>

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

rely on strong assumptions (e.g., well-conditioned Hessians, smoothness) and expensive Hessian-based updates, limiting their scalability to Deep Neural Networks (DNNs) and large forget sets. Specifically, the highly non-linear nature of DNNs often violates these preconditions, undermining the theoretical guarantees and practical performance. Meanwhile, Hessian computations (or equivalent approximations) incur heavy computational and memory overheads, scaling poorly with the size of model and forget set. Besides, these methods become inapplicable when retain data is unavailable.

In this paper, we propose **GUARD** (***G**radient-based **U**nlearning with **A**dversarial **R**obustness **D**efense*), a scalable and efficient framework that achieves unlearning while preserving retain-data adversarial robustness. Inheriting the intrinsic efficiency of GU paradigm, this plug-and-play solution is compatible with diverse GU loss functions, optimizers, data constraints, architectures and unlearning requests. GUARD is designed in response to two main causes of robustness degradation identified in our analysis: *Gradient Direction Conflict* and *Parameter Entanglement*. Specifically, we propose two main modules: (i) *Robustness-Aware Gradient Projection* suppresses gradient components that harm robustness; (ii) *Importance-Guided Parameter Selection* restricts updates to parameters critical for forgetting but insensitive to robustness. Both modules are underpinned by the construction of a compact *adversarially vulnerable dataset*, which ensures efficiency under varying data constraints—whether retain data is accessible or not. Our contributions are summarized as follows.

- We attribute the degradation of adversarial robustness in GU to two key factors: Gradient Direction Conflict and Parameter Entanglement.
- We propose GUARD, a scalable and efficient GU framework that preserves adversarial robustness on retain data while enabling effective unlearning.
- We validate GUARD via theoretical analysis and experiments on 6 settings. It outperforms 6 baselines with up to 6.3$\times$ higher robust accuracy, closely matching the Retrain oracle.

2 Related Work

Machine Unlearning. In MU, model manipulation is a typical type [49], including: (i) manual weight editing or pruning [16, 33, 47]; (ii) Hessian-based, second-order updates [9, 17, 19]; (iii) first-order optimization known as Gradient-based Unlearning (GU) [24, 29]. From the perspective of data usage, GU can be further categorized into methods that leverage both the forget and retain sets [10, 32, 45] and *retain-free* GU that updates the model using only the forget set [15, 40, 8, 7, 3, 11, 27]. Prior work has explored gradient-projection methods [21, 43] that make updates orthogonal to retain-data gradients, and importance-based methods [15, 6] that update only forget-critical parameters. However, such standard MU inadvertently degrades the model's adversarial robustness.

Adversarially Robust Fine-tuning. DNNs are vulnerable to adversarial examples [18], and adversarial training is a direct defense [37]. As models scale up, *robust fine-tuning* emerges to adapt pretrained models without compromising robustness. Studies [31, 22] compare fine-tuning strategies and find that robustness is better preserved by updating only a small parameter subset; in particular, fine-tuning only the classifier head suffices [42]. A line of work suggests that pruning helps preserve robustness by constraining model capacity, e.g., via structured [25], adaptive [1], or importance-based pruning [41]. Some further localize robustness-critical components [50] for selective updates, e.g., by estimating robust criticality [52] or via gradient-based importance [30]. Viewing MU as a form of fine-tuning, these inspire us to integrate adversarial defense directly into the optimization loop.

Machine Unlearning and Adversarial Attacks. Recent works exploring the interplay between MU and adversarial attacks fall into four lines. (i) Some incorporate adversarial perturbations to facilitate unlearning [8, 27, 26]. (ii) MU can also be used to enhance adversarial attacks [51]. (iii) Adversarial perturbations are combined with MU for broader objectives, e.g., to degrade retain generalization [23], or to unlearn backdoor triggers [48]. (iv) Studies [34, 35] explore MU on adversarially trained models. These highlight that integrating of privacy-centric unlearning with security-focused defense is vital for holistic model trustworthiness.

3 Preliminary and Problem Setup

3.1 Adversarial Robustness

Given a labeled dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ and a DNN f_θ trained with the empirical risk $\frac{1}{n}\sum_{i=1}^n \ell_{\mathrm{CE}}\big(f_\theta(x_i), y_i\big)$, where $\ell_{\mathrm{CE}}(\cdot,\cdot)$ is the cross-entropy loss. Given a clean input (x, y), an *Adversarial Example* refers to a perturbed version of x that is perceptually indistinguishable to humans yet induces erroneous model predictions. Specifically, we adopt the L_p threat model, where perturbations $\delta \in \mathbb{R}^d$ are constrained by $\|\delta\|_p \le \varepsilon$. Thus, the allowable set of adversarial examples is the L_p-ball around x: $B_p(x,\varepsilon) = \big\{x' \in \mathbb{R}^d : \|x' - x\|_p \le \varepsilon\big\}$. Adversarial examples are generated by iteratively applying Projected Gradient Descent (PGD): $x^{t+1} = \Pi_{B_p(x,\varepsilon)}\left(x^t + \alpha\, \nabla_x \ell_{\mathrm{CE}}(f_\theta(x^t), y)\right)$, where $t = 0, \ldots, T-1$ and α is the step size, Π is the projection operator. For brevity, we denote the adversarial example obtained after PGD by $x^{\mathrm{adv}} := \mathrm{PGD}_{\varepsilon,p}^{(T)}(x, y; \theta) = x^T$.

For each sample, the *adversarial loss* of model f_θ is $\ell_{\mathrm{adv}}(x, y; \theta) = \max_{\|\delta\|_p \le \varepsilon} \ell_{\mathrm{CE}}\big(f_\theta(x+\delta), y\big)$, which captures the worst-case loss under all admissible δ in the given threat model. On the entire $\mathcal{D}$, the *robust empirical risk* of f_θ is defined as: $\mathcal{L}_{\mathrm{adv}}(\mathcal{D}; \theta) = \frac{1}{|\mathcal{D}|}\sum_{x \in \mathcal{D}} \max_{\|\delta\|_p \le \varepsilon} \ell_{\mathrm{CE}}\big(f_\theta(x+\delta), y\big)$.

3.2 Robustness-Preserving Machine Unlearning

In image classification tasks, let $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ be the original training dataset. We use $\mathcal{D}_f = \{(x_i, y_i) \in \mathcal{D}\}$ to represent the forget set consisting of all samples to be forgotten, and $\mathcal{D}_r = \mathcal{D} \big/ \mathcal{D}_f$ to denote the retain set containing the information we want to preserve.

Let θ_{ori} be the original model obtained from either standard or adversarial training on $\mathcal{D}$. For any MU algorithm $\mathcal{U}$, we obtain the unlearned model $\theta_{\mathrm{unl}} = \mathcal{U}(\theta_{\mathrm{ori}})$. θ_{retr} denotes the model retrained from scratch using the same training pipeline on $\mathcal{D}_r$. In standard MU, the goal is to make θ_{unl} approximates θ_{retr} in output distribution, which is commonly assessed by the clean-accuracy gap on $\mathcal{D}_f$ and $\mathcal{D}_r$. Furthermore, let $\mathcal{P}_{\mathrm{adv}}$ denote the distribution of worst-case adversarial examples derived from the retain data distribution within the ℓ_p-ball of radius ε. We define $\mathcal{U}$ as *robustness-preserving* if it incurs a negligible gap in the expected risk over $\mathcal{P}_{\mathrm{adv}}$: $\Delta\mathcal{R}^* := \mathcal{R}^*(\theta_{\mathrm{unl}}) - \mathcal{R}^*(\theta_{\mathrm{ori}})$, where $\mathcal{R}^*(\theta) = \mathbb{E}_{z \sim \mathcal{P}_{\mathrm{adv}}}[\ell_{\mathrm{CE}}(z; \theta)]$. In practice, this is quantified by the empirical loss gap $\Delta\mathcal{L}_{\mathrm{adv}} := \mathcal{L}_{\mathrm{adv}}(\mathcal{D}_r; \theta_{\mathrm{unl}}) - \mathcal{L}_{\mathrm{adv}}(\mathcal{D}_r; \theta_{\mathrm{ori}})$, or equivalently, the adversarial accuracy drop on $\mathcal{D}_r$.

4 Attribution of Robustness Degradation

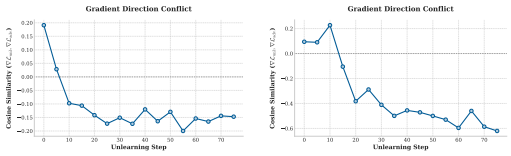
In this section, we analyse the key causes why GU degrades model's adversarial robustness. Consider the increase of retain-data adversarial loss during GU: $\Delta\mathcal{L}_{\mathrm{adv}} := \mathcal{L}_{\mathrm{adv}}(\mathcal{D}_r; \theta + \Delta\theta) - \mathcal{L}_{\mathrm{adv}}(\mathcal{D}_r; \theta)$, where θ is the current model parameters and $\Delta\theta$ is the GU update. We view $\Delta\mathcal{L}_{\mathrm{adv}}$ as a function of $\Delta\theta$. For GU using gradient descent to minimize an unlearning loss $\mathcal{L}_{\mathrm{unl}}$, we have $\Delta\theta = -\eta\nabla_\theta\mathcal{L}_{\mathrm{unl}}(\theta)$. Performing a second-order Taylor expansion of $\mathcal{L}_{\mathrm{adv}}(\mathcal{D}_r; \theta)$ around θ yields

$$\Delta\mathcal{L}_{\mathrm{adv}} \approx -\eta\big(\nabla_\theta\mathcal{L}_{\mathrm{adv}}^\top \nabla_\theta\mathcal{L}_{\mathrm{unl}}\big) + \frac{\eta^2}{2}\big(\nabla_\theta\mathcal{L}_{\mathrm{unl}}^\top H_{\mathrm{adv}} \nabla_\theta\mathcal{L}_{\mathrm{unl}}\big), \tag{1}$$

where $H_{\mathrm{adv}} = \nabla_\theta^2\mathcal{L}_{\mathrm{adv}}(\mathcal{D}_r; \theta)$ is the Hessian matrix. This mathematically decomposes robustness degradation into two distinct factors, which we term *Gradient Direction Conflict* and *Parameter Entanglement*, respectively.

Gradient Direction Conflict. It is revealed by the first-order linear term $-\eta(\nabla_\theta\mathcal{L}_{\mathrm{adv}}^\top \nabla_\theta\mathcal{L}_{\mathrm{unl}})$. This reflects the angle between the two gradients $\nabla_\theta\mathcal{L}_{\mathrm{adv}}$ and $\nabla_\theta\mathcal{L}_{\mathrm{unl}}$ in a high-dimensional space: when their dot product is negative, the directions conflict, meaning the unlearning update opposes the direction that decreases the adversarial loss and may therefore increase it.

Our gradient cosine-similarity curves in Fig. 1 empirically confirm this phenomenon. During GU, $\nabla_\theta\mathcal{L}_{\mathrm{unl}}$ is largely negatively aligned with $\nabla_\theta\mathcal{L}_{\mathrm{adv}}$, which becomes stronger as unlearning proceeds. We observe this trend consistently across different GU methods (additional results in Appendix D.2). This indicates an inherent directional conflict between the unlearning and robustness objectives.



(a) Gradient Ascent (GA) (b) Random Label (RL)

Figure 1: Gradients cosine similarity with All-CNN on CIFAR-10.

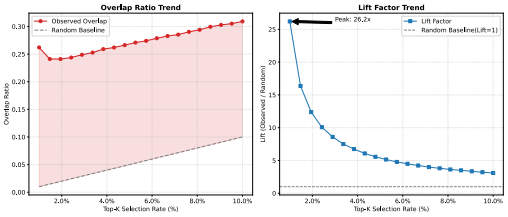


Figure 2: Top-$k\%$ Parameter Overlap with All-CNN on CIFAR-10.

Parameter Entanglement. It is revealed by the term $\frac{1}{2}\eta^2(\nabla_\theta \mathcal{L}_{\text{unl}}^\top H_{\text{adv}} \nabla_\theta \mathcal{L}_{\text{unl}})$. This second-order quadratic term captures curvature-induced robustness damage and is positive semidefinite. Approximating the Hessian by the empirical diagonal Fisher Information Matrix (FIM), we obtain $\frac{\eta^2}{2}\sum_j \mathbb{E}_{\mathcal{D}_r}\left[\left(\nabla_{\theta_j}\ell_{\text{adv}}\right)^2\right]\left(\nabla_{\theta_j}\mathcal{L}_{\text{unl}}\right)^2$, where j indexes parameters. It shows that robustness degrades when many parameters simultaneously exhibit large gradients for both losses. In other words, updating parameters critical to both $\mathcal{L}_{\text{unl}}$ and $\mathcal{L}_{\text{adv}}$ can quadratically increase the adversarial loss.

We empirically confirm this factor. Ranking parameters according to their average gradient magnitudes for $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{unl}}$, we measure the overlap between their Top-$k\%$ sets. Statistically, if the two objectives are unrelated, their gradient distributions should be independent, the expected overlap would be $k\%$. We therefore define a *lift factor* as the ratio of the observed overlap to this expectation. Fig. 2 shows: the probability that unlearning-critical parameters (Top-$k\%$ by $\nabla_\theta \mathcal{L}_{\text{unl}}$) are also robustness-critical (Top-$k\%$ by $\nabla_\theta \mathcal{L}_{\text{adv}}$) is significantly higher than the baseline, and it rises as k decreases. This confirms a strong Parameter Entanglement between the two objectives, particularly in high-importance regions. Consequently, large unlearning updates tend to act on high-curvature directions, amplifying the second-order term and degrading robustness.

5 Method

Based on our attribution, we propose GUARD accordingly, with Robustness-Aware Gradient Projection and Importance-Guided Parameter Selection explicitly tackling the 2 factors. Both modules rely on a small adversarially vulnerable dataset $\mathcal{D}_{\text{v}}$, constructed from retain data $\mathcal{D}_r$ when available, or—more challengingly—from the forget data $\mathcal{D}_f$ when $\mathcal{D}_r$ is inaccessible.

5.1 Adversarially Vulnerable Dataset Construction

Geometrically, adversarial robustness hinges primarily on samples near decision boundaries, where predictions rely heavily on non-robust features. Thus, we construct a small yet robustness-critical *adversarially vulnerable set* $\mathcal{D}_v$ ($|\mathcal{D}_v| \ll |\mathcal{D}_r|$). By prioritizing robustness preservation on this worst-case set rather than the entire $\mathcal{D}_r$, we significantly reduce computational and storage overhead.

5.1.1 When Retain Data is Available

When the retain set $\mathcal{D}_r$ is (partially) accessible, we construct $\mathcal{D}_v$ by applying moderately adversarial perturbations to a randomly sampled subset $\mathcal{D}_r^{\text{sub}} \subset \mathcal{D}_r$ of size m: $\tilde{x} = \text{PGD}_{\varepsilon_1,p}^{(T)}(x, y; \theta), \forall (x, y) \in \mathcal{D}_r^{\text{sub}}$. To ensure $\tilde{x}$ approaches the decision boundary without crossing it, we use a small perturbation radius ε_1 and iteration number T. Also, we filter for samples that remain correctly classified as $f_\theta(\tilde{x}) = y$. The resulting pairs $(\tilde{x}, y)$ form the adversarially vulnerable set $\mathcal{D}_v$.

Geometrically, such boundary-proximal samples in $\mathcal{D}_v$ possess small adversarial margins and weak robust features. A small additional perturbation, or a slight parameter change during unlearning, is enough to flip their predictions. Thus, they represent the adversarial vulnerability of $\mathcal{D}_r$.

5.1.2 When Retain Data is Unavailable

In practical scenarios when $\mathcal{D}_r$ is not accessible, constructing the adversarially vulnerable set to quantify adversarial robustness for $\mathcal{D}_r$ is challenging. This is because $\mathcal{D}_f$ and $\mathcal{D}_r$ typically differ

significantly in distribution. Particularly in class-wise unlearning, their label sets are even disjoint, so we can access no samples—and thus no feature information—from the retain classes.

To address this, we propose to *simulate* the adversarially vulnerable set of the retain data using $\mathcal{D}_f$. First, to narrow the domain gap between $\mathcal{D}_f$ and $\mathcal{D}_r$, we apply a specific set of strong semantic augmentations $\mathcal{T}(\cdot)$ (e.g., random cropping, patch recomposition) to each $(x^f, y^f) \in \mathcal{D}_f$ and obtain $\hat{x} = \mathcal{T}(x^f)$. This serves to attenuate the $\mathcal{D}_f$ samples of robust semantic features associated with y^f, rendering non-robust, gradient-sensitive components more dominant within the samples. Thus, the augmentation acts as a coarse perturbation, making the samples easier to steer toward retain classes.

Next, we apply targeted PGD on $\hat{x}$ toward a random retain label y^r: $\tilde{x} = \mathrm{PGD}^{(T)}_{\varepsilon_2,p}(\hat{x}, y^r; \theta)$. We form the simulated $\mathcal{D}_v$ using perturbed samples that satisfy $f_\theta(\tilde{x}) = y^r$. This process pushes $\mathcal{D}_f$ samples into the decision regions of the retain class, encoding the model's learned knowledge. As they lack strong robust features and exhibit high gradient sensitivity, these samples reside near the decision boundaries of y^r, serving as a suitable proxy for the true adversarially vulnerable set.

We simulate the $\mathcal{D}_v$ set based on $\mathcal{D}_f$, rather than training a generative model or optimizing over pure noise, primarily for distributional alignment and efficiency: First, $\mathcal{D}_f$ and $\mathcal{D}_r$ originate from the same domain and share similar semantic structure. Using $\mathcal{D}_f$ as a seed allows $\mathcal{D}_v$ to inherit this distributional prior and stay on (or close to) the model's learned data manifold. Second, since we aim to capture model-centric vulnerability rather than perceptual fidelity, moderate perturbations to existing samples suffice, avoiding costly generative modeling.

5.2 Robustness-Aware Gradient Projection

In response to the Gradient Direction Conflict, we treat $\mathcal{D}_v = \{(x_i^v, y_i^v)\}_{i=1}^m$ as anchors, and suppress the robustness-harmful directions of the unlearning gradients at these anchors. For efficiency, we may sample a small subset from $\mathcal{D}_v$, and resample it periodically (which we term as 'refresh').

For each anchor (x_i^v, y_i^v), we define its *robustness indicator* as $r_i(\theta) \coloneqq \ell_{\mathrm{CE}}\big(f_\theta(x_i^v), y_i^v\big)$. Then, the empirical mean of the robustness indicator is $\mathcal{R}(\theta) \coloneqq \frac{1}{m}\sum_{i=1}^m r_i(\theta) = \frac{1}{m}\sum_{i=1}^m \ell_{\mathrm{CE}}\big(f_\theta(x_i^v), y_i^v\big)$. Since $\mathcal{D}_v$ explicitly captures the vulnerable regions near robust decision boundaries, $r_i(\theta)$ and $R(\theta)$ serve as proxies for the adversarial loss $\ell_{\mathrm{adv}}(x_i^r, y_i^r; \theta)$ and the robust empirical risk $\mathcal{L}_{\mathrm{adv}}(\mathcal{D}_r; \theta)$, respectively. That is, $\big|\mathcal{L}_{\mathrm{adv}}(\mathcal{D}_r; \theta) - \mathcal{R}(\mathcal{D}_v; \theta)\big| \le \delta$, where $\delta > 0$ is a small margin.

With a small step size, the r_i change is approximated by a first-order Taylor expansion: $\Delta r_i(\theta) \approx \nabla_\theta r_i(\theta)^\top \Delta\theta$. Consider a general update $\Delta\theta = \eta p$, where $\eta > 0$ is the learning rate and $p = -P\nabla_\theta \mathcal{L}_{\mathrm{unl}}(\theta)$ is the update direction mediated by a first-order preconditioner P (e.g., $P = I$ for SGD). Absorbing η into a tolerance κ_i, we constrain the robustness change $\Delta r_i(\theta)$ at each anchor:

$$\nabla_\theta r_i(\theta)^\top p \le \kappa_i, \qquad \kappa_i \ge 0. \tag{2}$$

Stacking the gradients of all m anchors as the Jacobian matrix $J \in \mathbb{R}^{m\times d}$ with the i-th row $J_i^\top = \nabla_\theta r_i(\theta)^\top$, the constraints in Eq. (2) define a feasible region $Jq \le \kappa$, where $\kappa = (\kappa_1, \ldots, \kappa_m)^\top$. To ensure anchors' robustness, we project the original unlearning direction p onto this region. This yields a Quadratic Projection (QP) problem that seeks the optimal direction q by minimizing its deviation from p under the P^{-1} norm:

$$\min_q \ \frac{1}{2}\|q - p\|_{P^{-1}}^2 \quad \text{s.t.} \quad Jq \le \kappa, \tag{3}$$

where $\|v\|_{P^{-1}}^2 := v^\top P^{-1} v$. In other words, among all directions that maintain the robustness levels, q is geometrically closest to the original unlearning direction p.

By forming the Lagrangian, the KKT stationarity condition yields the update structure $q = p - PJ^\top\alpha$. We estimate the active set by applying the positive-part operator $[\cdot]_+$ to the initial violations (where $Jp - \kappa > 0$). This allows us to approximate the inequalities as a linear system $(JPJ^\top)\alpha = [Jp - \kappa]_+$, which leads to the closed-form solution of the optimal update direction q:

$$q = p - PJ^\top(JPJ^\top)^{-1}[Jp - \kappa]_+. \tag{4}$$

When $\kappa = 0$, we have $q = \big(I - PJ^\top(JPJ^\top)^{-1}J\big)\,p$, where $I - PJ^\top(JPJ^\top)^{-1}J$ can be viewed as the P^{-1}-orthogonal projection operator onto the linear subspace $\{q : Jq = 0\}$.

JL approximation. The cost of computing $S = JPJ^\top$ and Jp ($J \in \mathbb{R}^{m\times d}$) is dominated by the large parameter dimension d. We thus apply a Johnson–Lindenstrauss (JL) projection for approximation. We draw a random matrix $\Phi \in \mathbb{R}^{k\times d}$ with $k \ll d$ and compute $\tilde{v}_i = \Phi P^{1/2}\nabla_\theta r_i(\theta)$, $\tilde{u} = \Phi P^{-1/2}p$. We then replace S and Jp with the JL-approximated version:

$$S_{ij} = \nabla_\theta r_i(\theta)^\top P \nabla_\theta r_j(\theta) \approx \tilde{v}_i^\top \tilde{v}_j, \quad (Jp)_i = \nabla_\theta r_i(\theta)^\top p \approx \tilde{v}_i^\top \tilde{u}. \tag{5}$$

For the construction of S, this reduces the computation from $\mathcal{O}(m^2 d)$ to $\mathcal{O}(m^2 k)$ and storage from $\mathcal{O}(md)$ to $\mathcal{O}(mk)$, where $k = \mathcal{O}(\beta^{-2}\log m)$ is independent of d (tolerance $\beta \in (0,1)$). Further, we use a streaming JL to avoid storing the full $k \times d$ matrix Φ. See Appendix B.4 for details.

5.3 Importance-Guided Parameter Selection

To mitigate the Parameter Entanglement, we introduce Importance-Guided Parameter Selection. Recalling the term $\frac{1}{2}\eta^2 \sum_i (\nabla_{\theta_i}\mathcal{L}_{\text{adv}} \nabla_{\theta_i}\mathcal{L}_{\text{unl}})^2$ in Section 4, achieving unlearning while preserving robustness requires restricting the optimization to parameters that exhibit large unlearning gradients yet small adversarial gradients. Since the gradient magnitude reflects the sensitivity of the task risk to parameter changes, we use it to quantify importance. Given a task using loss ℓ and dataset $\mathcal{D}$, for the k-th component θ_k of the model parameters $\theta \in \mathbb{R}^d$, we define the importance as:

$$I_k(\mathcal{D}) = \tfrac{1}{|\mathcal{D}|}\sum\nolimits_{(x,y)\in\mathcal{D}} |\nabla_{\theta_k}\ell(f_\theta(x), y)|. \tag{6}$$

We instantiate this metric for both objectives. For the forgetting objective, we adopt gradient ascent on $\mathcal{D}_f$ (i.e., $\ell_{\text{unl}} = -\ell_{\text{CE}}$), defining the forgetting importance as: $I_k^{(\text{f})} = I_k(\mathcal{D}_f) = \frac{1}{|\mathcal{D}_f|}\sum_{(x,y)\in\mathcal{D}_f} |\nabla_{\theta_k}\ell_{\text{CE}}(f_\theta(x), y)|$. For robustness, based on the indicator $r_i(\theta) = \ell_{\text{CE}}$, we define the robustness importance as: $I_k^{(\text{adv})} = I_k(\mathcal{D}_v) = \frac{1}{|\mathcal{D}_v|}\sum_{(x,y)\in\mathcal{D}_v} |\nabla_{\theta_k}\ell_{\text{CE}}(f_\theta(x), y)|$. For each model parameter, we compute a ratio score based on its importance:

$$s_k = I_k^{(\text{f})} / (I_k^{(\text{adv})} + \epsilon), \tag{7}$$

where $\epsilon > 0$ avoids division by zero. Then, we construct a binary mask $M \in \{0,1\}^d$, where the k-th entry is $M_k = \mathbb{I}\left[s_k \in \text{TopK}(s,\rho)\right]$, where $\text{TopK}(\cdot,\rho)$ selects the top ρ percentile. During unlearning, this mask is applied to the projected direction q to get the parameter update $\Delta\theta = \eta\,(M \odot q)$, where $\odot$ is element-wise multiplication. Consequently, we update only the parameters with large s—deemed *critical for forgetting yet insensitive to robustness*—while keeping the rest unchanged.

6 Theoretical Analysis

This section provides theoretical guarantees (all proofs in Appendix C). First, in Section 6.1, we bridge the gap between our empirical robustness indicator and the true expected risk. Further, we demonstrate in Sections 6.2 and 6.3 how our two core modules strictly bound the degradation of this empirical robustness compared to standard unlearning baselines.

6.1 Analysis of Adversarially Vulnerable Dataset

Since we aim to preserve the true expected risk $\mathcal{R}^*(\theta)$ via the empirical robustness indicator $\mathcal{R}(\theta)$, we must formalize their relationship. To quantify this gap, we adapt the uniform convergence bound based on Rademacher complexity [38, 2]. We assume that $\ell_{\text{CE}} \in [0, M]$, and $\mathcal{D}_v$ consists of i.i.d. samples drawn from the worst-case adversarial distribution $\mathcal{P}_{\text{adv}}$. Let $\mathcal{F}$ be the loss function class, it holds with probability at least $1-\delta$ that $\sup_{\theta\in\Theta}|\mathcal{R}^*(\theta) - \mathcal{R}(\theta)| \le 2\mathfrak{R}_m(\mathcal{F}) + 3M\sqrt{\log(2/\delta)/(2m)} := \epsilon_{\text{gen}}$, where $m = |\mathcal{D}_v|$ and $\mathfrak{R}_m(\mathcal{F})$ is the empirical Rademacher complexity, ϵ_{gen} is the error bound.

To formalize the degradation during unlearning, let $\Delta\mathcal{R} = \mathcal{R}(\theta_{\text{unl}}) - \mathcal{R}(\theta_{\text{ori}})$ and $\Delta\mathcal{R}^* = \mathcal{R}^*(\theta_{\text{unl}}) - \mathcal{R}^*(\theta_{\text{ori}})$ denote the empirical and true robustness degradation, respectively. Applying the uniform convergence bound to both θ_{unl} and θ_{ori}, we can bound the true degradation as:

$$\Delta\mathcal{R}^* \le \big(\mathcal{R}(\theta_{\text{unl}}) + \epsilon_{\text{gen}}\big) - \big(\mathcal{R}(\theta_{\text{ori}}) - \epsilon_{\text{gen}}\big) = \Delta\mathcal{R} + 2\epsilon_{\text{gen}}. \tag{8}$$

This directly links our empirical objective to the theoretical goal, ensuring that constraining $\Delta\mathcal{R}$ strictly limits the true robustness degradation $\Delta\mathcal{R}^*$ up to the generalization error $2\epsilon_{\text{gen}}$.

6.2 Analysis of Robustness-Aware Gradient Projection

We now derive the upper bound on $\Delta\mathcal{R}$ for our Gradient Projection strategy, in Theorem 6.2.

Assumption 6.1. (1) Smoothness: The robust indicator $r_i(\theta)$, the empirical risk $\mathcal{R}(\theta)$, and the unlearning loss $\mathcal{L}_{\text{unl}}(\theta)$ are Lipschitz smooth with constants L_i, L_R, and L_F. **(2) Boundedness:** The unlearning gradient is bounded by $\|\nabla\mathcal{L}_{\text{unl}}(\theta)\|_2 \leq G$, and the preconditioner P is positive definite with bounded maximum eigenvalue $\lambda_{\max}(P)$.

Theorem 6.2 (Robustness Degradation Bound via Gradient Projection)**.** *Consider a T-step GU with Robustness-Aware Gradient Projection, updating θ_{ori} to $\theta_{\text{unl}}^{\text{proj}}$. Let $\Delta R_{\text{proj}} \coloneqq \mathcal{R}(\theta_{\text{unl}}^{\text{proj}}) - \mathcal{R}(\theta_{\text{ori}})$ denote the increment of empirical robustness indicator. Under Assumption 6.1, we have:*

$$\Delta R_{\text{proj}} \leq \text{Bound}(\Delta R_{\text{proj}}) \coloneqq T\left(\eta\kappa + \frac{\bar{L}}{2}\eta^2 B^2\right) \tag{9}$$

where $\bar{L} \coloneqq \max_i L_i$ is the smoothness constant, and $B = \lambda_{\max}(P)G$ is the upper bound on the norm of the projected update direction $\|q_t\|$.

Proposition 6.3 compares this with the baseline (standard GU without robustness constraints), whose degradation is bounded by $\Delta R_{\text{base}} \leq \text{Bound}(\Delta R_{\text{base}}) \coloneqq T \cdot (\eta\bar{\mathcal{A}}_{\text{base}} + \frac{\bar{L}}{2}\eta^2 B^2)$, where $\bar{\mathcal{A}}_{\text{base}}$ is the average magnitude of Gradient Direction Conflict: $\bar{\mathcal{A}}_{\text{base}} \coloneqq \frac{1}{T}\sum_{t=0}^{T-1}(\frac{1}{m}\sum_{i=1}^{m}\nabla r_i(\theta_t)^\top p_t)$.

Proposition 6.3 (Tighter Bound via Gradient Projection)**.** *By setting the tolerance κ below the baseline's average gradient conflict ($\kappa < \bar{\mathcal{A}}_{\text{base}}$), our method guarantees a strictly tighter upper bound on robustness degradation:* $\text{Bound}(\Delta R_{\text{proj}}) < \text{Bound}(\Delta R_{\text{base}})$.

Remark 6.4. Condition $\kappa < \bar{\mathcal{A}}_{\text{base}}$ is readily satisfied. Since standard GU inherently conflicts with robustness ($\bar{\mathcal{A}}_{\text{base}} > 0$), setting a small hyperparameter κ (e.g. $\kappa = 0$) guarantees the tighter bound.

6.3 Analysis of Importance-Guided Parameter Selection

We then derive the upper bound on $\Delta\mathcal{R}$ for our Parameter Selection strategy, in Theorem 6.5.

Theorem 6.5 (Robustness Degradation Bound via Parameter Selection)**.** *Consider a T-step GU with Importance-Guided Parameter Selection, updating θ_{ori} to $\theta_{\text{unl}}^{\text{sel}}$. Let $\Delta R_{\text{sel}} \coloneqq \mathcal{R}(\theta_{\text{unl}}^{\text{sel}}) - \mathcal{R}(\theta_{\text{ori}})$ denote the increment in the empirical robustness indicator. Let $\Delta\mathcal{L}_{\text{unl}} = \mathcal{L}_{\text{unl}}(\theta_{\text{ori}}) - \mathcal{L}_{\text{unl}}(\theta_{\text{unl}}^{\text{sel}})$ be the total change in unlearning loss. Under Assumption 6.1, we have:*

$$\Delta\mathcal{R}_{\text{sel}} \leq \text{Bound}(\Delta\mathcal{R}_{\text{sel}}) \coloneqq \frac{1}{\tau_{\text{sel}}}\Delta\mathcal{L}_{\text{unl}} + T \cdot C_{\text{sel}} \cdot (\rho\eta^2 B^2) \tag{10}$$

where τ_{sel} is the importance threshold determined by ρ, $C_{\text{sel}} \coloneqq \frac{L_R}{2} + \frac{L_F}{2\tau_{\text{sel}}}$ captures the smoothness of both objectives, and $B = \lambda_{\max}(P)G$ is the upper norm bound on update direction $\|q_t\|$.

Proposition 6.6 compares this to the standard GU baseline, bounded by $\text{Bound}(\Delta\mathcal{R}_{\text{base}}) \coloneqq \frac{1}{\tau_{\min}}\mathcal{L}_{\text{unl}} + T \cdot C_{\text{base}} \cdot \eta^2 B^2$, where $C_{\text{base}} \coloneqq \frac{L_R}{2} + \frac{L_F}{2\tau_{\min}}$, $\tau_{\min}$ is the minimum importance score.

Proposition 6.6 (Tighter Bound via Parameter Selection)**.** *As the threshold τ_{sel} strictly exceeds the baseline's worst-case $\tau_{\min}$(implying $\rho < 1$), our method inherently guarantees a strictly tighter robust degradation upper bound:* $\text{Bound}(\Delta\mathcal{R}_{\text{sel}}) < \text{Bound}(\Delta\mathcal{R}_{\text{base}})$.

7 Experiments

7.1 Experimental Settings

The experiments are implemented with 1 NVIDIA GeForce RTX 4090, using 3 datasets: Tiny ImageNet [13], CIFAR-10 and CIFAR-100 [28]. For each dataset, we adopt 2 DNNs: AllCNN [44] and ResNet18 [20]. We also evaluate ViT [14] with CIFAR-10. Appendix D details the settings and additional experiments, including MIA evaluations, multi-class unlearning, visualizations and etc.

Baselines include: (1) **Original** (2) **Retrain** (3) **Gradient Ascent (GA)** [17] (4) **Boundary Shrink (BS)** [8] (5) **Saliency Unlearning (SalUn)** [15] (6) **Fine-tuning(FT)** [17] (7) **UNSIR** [45] (8)

Table 1: Class unlearning results on CIFAR-10 (target class: 'Cat').

	Methods	AllCNN					ResNet-18				
	Metrics	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\mathrm{RA}_r \uparrow$	Avg. Gap	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\mathrm{RA}_r \uparrow$	Avg. Gap
	Original	$68.80_{\pm 0.52}$ (68.80)	$91.62_{\pm 0.14}$ (3.52)	$81.00_{\pm 0.33}$ (1.41)	$71.45_{\pm 0.29}$ (2.37)	19.03	$100.0_{\pm 0.00}$ (100.0)	$100.0_{\pm 0.00}$ (0.00)	$79.00_{\pm 0.20}$ (3.89)	$99.96_{\pm 0.02}$ (0.04)	25.98
	Retrain	$0.00_{\pm 0.00}$ (0.00)	$95.14_{\pm 0.12}$ (0.00)	$82.41_{\pm 0.21}$ (0.00)	$73.82_{\pm 0.45}$ (0.00)	0.00	$0.00_{\pm 0.00}$ (0.00)	$100.0_{\pm 0.00}$ (0.00)	$82.89_{\pm 0.08}$ (0.00)	$100.0_{\pm 0.00}$ (0.00)	0.00
	Muter	$9.80_{\pm 0.64}$ (9.80)	$53.89_{\pm 1.23}$ (41.25)	$46.36_{\pm 0.88}$ (36.05)	$30.30_{\pm 1.05}$ (43.52)	32.66	$6.80_{\pm 0.20}$ (6.80)	$99.99_{\pm 0.00}$ (0.01)	$77.98_{\pm 0.02}$ (4.91)	$98.81_{\pm 0.01}$ (1.19)	3.23
w/o D_r	GA	$0.39_{\pm 0.05}$ (0.39)	$84.28_{\pm 0.44}$ (10.86)	$75.87_{\pm 0.62}$ (6.54)	$55.86_{\pm 1.12}$ (17.96)	8.94	$0.18_{\pm 0.03}$ (0.18)	$98.64_{\pm 0.55}$ (1.36)	$74.19_{\pm 0.76}$ (8.70)	$81.68_{\pm 0.29}$ (18.32)	7.14
w/o D_r	**GA+ours**	$0.76_{\pm 0.11}$ (0.76)	$92.60_{\pm 0.31}$ (2.54)	$81.91_{\pm 0.25}$ (0.50)	$75.12_{\pm 0.58}$ (-1.30)	0.63	$0.00_{\pm 0.00}$ (0.00)	$100.0_{\pm 0.00}$ (0.00)	$81.20_{\pm 0.04}$ (1.69)	$99.96_{\pm 0.01}$ (0.04)	**0.43**
w/o D_r	BS	$0.42_{\pm 0.09}$ (0.42)	$81.60_{\pm 0.72}$ (13.54)	$72.32_{\pm 0.55}$ (10.09)	$57.37_{\pm 1.34}$ (16.45)	10.13	$2.02_{\pm 0.28}$ (2.02)	$99.19_{\pm 0.09}$ (0.81)	$75.02_{\pm 0.33}$ (7.87)	$78.51_{\pm 0.52}$ (21.49)	8.05
w/o D_r	**BS+ours**	$0.52_{\pm 0.08}$ (0.52)	$92.46_{\pm 0.19}$ (2.68)	$81.88_{\pm 0.24}$ (0.53)	$74.71_{\pm 0.41}$ (-0.89)	0.71	$1.94_{\pm 0.35}$ (1.94)	$100.0_{\pm 0.00}$ (0.00)	$79.89_{\pm 0.58}$ (3.00)	$99.14_{\pm 0.15}$ (0.86)	1.45
w/o D_r	SalUn	$0.46_{\pm 0.07}$ (0.46)	$86.38_{\pm 0.51}$ (8.76)	$77.09_{\pm 0.48}$ (5.32)	$59.91_{\pm 0.92}$ (13.91)	7.11	$0.82_{\pm 0.14}$ (0.82)	$100.0_{\pm 0.00}$ (0.00)	$78.58_{\pm 0.29}$ (4.31)	$95.66_{\pm 0.22}$ (4.34)	2.37
w/o D_r	**SalUn+ours**	$0.54_{\pm 0.10}$ (0.54)	$92.19_{\pm 0.26}$ (2.95)	$81.39_{\pm 0.33}$ (1.02)	$74.36_{\pm 0.47}$ (-0.54)	0.99	$1.09_{\pm 0.21}$ (1.09)	$100.0_{\pm 0.00}$ (0.00)	$80.99_{\pm 0.32}$ (1.90)	$99.86_{\pm 0.02}$ (0.14)	0.78
w/ D_r	FT	$0.00_{\pm 0.00}$ (0.00)	$95.70_{\pm 0.13}$ (-0.56)	$85.11_{\pm 0.29}$ (-2.70)	$49.16_{\pm 0.15}$ (24.66)	5.35	$0.00_{\pm 0.00}$ (0.00)	$97.18_{\pm 0.24}$ (2.82)	$83.41_{\pm 0.45}$ (-0.52)	$11.80_{\pm 0.92}$ (88.20)	22.63
w/ D_r	**FT+ours**	$0.30_{\pm 0.04}$ (0.30)	$94.60_{\pm 0.24}$ (0.54)	$84.47_{\pm 0.45}$ (-2.06)	$74.43_{\pm 0.33}$ (-0.61)	**-0.46**	$0.66_{\pm 0.16}$ (0.66)	$99.89_{\pm 0.04}$ (0.11)	$78.30_{\pm 0.55}$ (4.59)	$74.30_{\pm 0.61}$ (25.70)	7.77
w/ D_r	UNSIR	$0.00_{\pm 0.00}$ (0.00)	$81.03_{\pm 0.84}$ (14.11)	$77.28_{\pm 0.76}$ (5.13)	$15.06_{\pm 1.02}$ (58.76)	19.50	$0.02_{\pm 0.01}$ (0.02)	$94.63_{\pm 0.39}$ (5.37)	$76.06_{\pm 0.81}$ (6.83)	$55.41_{\pm 1.18}$ (44.59)	14.20
w/ D_r	**UNSIR+ours**	$0.00_{\pm 0.00}$ (0.00)	$93.27_{\pm 0.22}$ (1.87)	$82.52_{\pm 0.27}$ (-0.11)	$74.45_{\pm 0.43}$ (-0.63)	0.28	$0.07_{\pm 0.02}$ (0.07)	$99.98_{\pm 0.00}$ (0.02)	$78.86_{\pm 0.04}$ (4.03)	$98.59_{\pm 0.01}$ (1.41)	1.38

Table 2: Random unlearn on CIFAR10, AllCNN.

	Methods	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\mathrm{RA}_r \uparrow$	Avg. Gap
	Original	$89.12_{\pm 0.07}$ (5.78)	$89.36_{\pm 0.10}$ (-2.09)	$78.11_{\pm 0.15}$ (-3.13)	$67.69_{\pm 0.21}$ (-1.55)	-0.25
	Retrain	$83.34_{\pm 0.16}$ (0.00)	$87.27_{\pm 0.15}$ (0.00)	$74.98_{\pm 0.22}$ (0.00)	$66.14_{\pm 0.18}$ (0.00)	0.00
	Muter	$88.60_{\pm 0.22}$ (5.26)	$89.01_{\pm 0.79}$ (-1.74)	$75.83_{\pm 0.50}$ (-0.85)	$66.08_{\pm 0.08}$ (0.06)	0.68
w/o D_r	GA	$85.36_{\pm 0.34}$ (2.02)	$86.41_{\pm 0.51}$ (0.86)	$74.80_{\pm 0.28}$ (0.18)	$64.46_{\pm 0.62}$ (1.68)	1.19
w/o D_r	**GA+ours**	$86.06_{\pm 0.19}$ (2.72)	$87.19_{\pm 0.33}$ (0.08)	$75.69_{\pm 0.41}$ (-0.71)	$66.71_{\pm 0.55}$ (-0.57)	0.38
w/o D_r	BS	$85.72_{\pm 0.45}$ (2.38)	$86.53_{\pm 0.27}$ (0.74)	$74.24_{\pm 0.64}$ (0.74)	$61.12_{\pm 0.71}$ (5.02)	2.22
w/o D_r	**BS+ours**	$85.69_{\pm 0.22}$ (2.35)	$86.94_{\pm 0.16}$ (0.33)	$75.32_{\pm 0.39}$ (-0.34)	$65.86_{\pm 0.48}$ (0.28)	0.66
w/o D_r	SalUn	$85.66_{\pm 0.53}$ (2.32)	$86.72_{\pm 0.38}$ (0.55)	$75.32_{\pm 0.42}$ (-0.34)	$64.47_{\pm 0.67}$ (1.67)	1.05
w/o D_r	**SalUn+ours**	$84.92_{\pm 0.29}$ (1.58)	$86.96_{\pm 0.14}$ (0.31)	$76.58_{\pm 0.35}$ (-1.60)	$66.88_{\pm 0.52}$ (-0.74)	**-0.11**
w/ D_r	FT	$89.18_{\pm 0.61}$ (5.84)	$98.57_{\pm 0.23}$ (-11.30)	$81.52_{\pm 0.46}$ (-6.54)	$27.06_{\pm 0.82}$ (39.08)	6.77
w/ D_r	**FT+ours**	$88.70_{\pm 0.06}$ (5.36)	$91.05_{\pm 0.10}$ (-3.78)	$78.10_{\pm 0.06}$ (-3.12)	$61.24_{\pm 0.08}$ (4.90)	0.84
w/ D_r	UNSIR	$83.96_{\pm 0.35}$ (0.62)	$87.11_{\pm 0.29}$ (0.16)	$79.93_{\pm 0.12}$ (-4.95)	$21.28_{\pm 0.10}$ (44.86)	10.17
w/ D_r	**UNSIR+ours**	$84.88_{\pm 0.37}$ (1.54)	$87.08_{\pm 0.25}$ (0.19)	$75.78_{\pm 0.43}$ (-0.80)	$65.87_{\pm 0.59}$ (0.27)	0.30

Table 3: Ablation on CIFAR-10 using GA.

	Module		Metrics				
	Proj	Sel	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\mathrm{RA}_r \uparrow$	Avg. Gap
AllCNN	Retrain		$0.00_{\pm 0.00}$ (0.00)	$95.14_{\pm 0.12}$ (0.00)	$82.41_{\pm 0.21}$ (0.00)	$73.82_{\pm 0.45}$ (0.00)	0.00
AllCNN	-	-	$0.39_{\pm 0.05}$ (0.39)	$84.28_{\pm 0.44}$ (10.86)	$75.87_{\pm 0.62}$ (6.54)	$55.86_{\pm 1.12}$ (17.96)	8.94
AllCNN	✓	-	$1.08_{\pm 0.19}$ (1.08)	$92.70_{\pm 0.31}$ (2.44)	$82.12_{\pm 0.28}$ (0.29)	$70.06_{\pm 0.57}$ (3.76)	1.89
AllCNN	-	✓	$0.00_{\pm 0.00}$ (0.00)	$91.45_{\pm 0.38}$ (3.69)	$80.24_{\pm 0.42}$ (2.17)	$71.15_{\pm 0.66}$ (2.67)	2.13
AllCNN	✓	✓	$0.76_{\pm 0.11}$ (0.76)	$92.60_{\pm 0.31}$ (2.54)	$81.91_{\pm 0.25}$ (0.50)	$75.12_{\pm 0.58}$ (-1.30)	0.63
ResNet18	Retrain		$0.00_{\pm 0.00}$ (0.00)	$100.0_{\pm 0.00}$ (0.00)	$82.89_{\pm 0.18}$ (0.00)	$100.0_{\pm 0.00}$ (0.00)	0.00
ResNet18	-	-	$0.18_{\pm 0.06}$ (0.18)	$98.64_{\pm 0.15}$ (1.36)	$74.19_{\pm 0.55}$ (8.70)	$81.68_{\pm 0.89}$ (18.32)	7.14
ResNet18	✓	-	$0.48_{\pm 0.12}$ (0.48)	$99.80_{\pm 0.09}$ (0.20)	$77.84_{\pm 0.37}$ (5.05)	$94.40_{\pm 0.24}$ (5.60)	2.83
ResNet18	-	✓	$0.00_{\pm 0.00}$ (0.00)	$99.94_{\pm 0.05}$ (0.06)	$79.28_{\pm 0.41}$ (3.61)	$97.46_{\pm 0.33}$ (2.54)	1.55
ResNet18	✓	✓	$0.00_{\pm 0.00}$ (0.00)	$100.0_{\pm 0.00}$ (0.00)	$81.20_{\pm 0.41}$ (1.69)	$99.96_{\pm 0.01}$ (0.04)	0.43

Muter [35]. (3)–(5) are retain-free GU using only forget data, while (6) and (7) are GU relying on retain data. **Original** represents the trained model to be forgotten and **Retrain** is the retrain-from-scratch oracle, both obtained via adversarial training under identical settings. We adopt FGSM [18] with perturbation radius $\epsilon = 0.02$ as the threat model for adversarial training and robust evaluation.

To evaluate unlearning, we use clean accuracy on the train retain set (A_r), train forget set (A_f), and test set (A_t). To evaluate retain adversarial robustness, we use robust accuracy RA_r on adversarial examples of the train retain data. To evaluate efficiency, we record the running time in seconds. We report the mean and standard deviation averaged over 5 runs. We calculate the performance gap for each metric relative to the Retrain oracle, where a negative gap indicates better performance and vice versa. 'Avg. Gap' is the mean gap across the 4 metrics. For fairness, we tuned all baselines to align their unlearning performance to the Retrain oracle before evaluating their robustness.

7.2 Class Unlearning

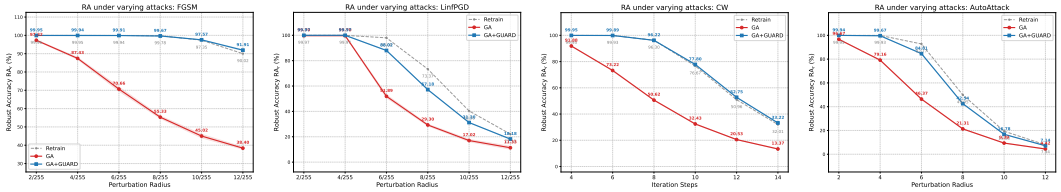
We first evaluate class unlearning, where the label sets for retain and forget data are strictly disjoint. The goal is to minimize forget-class accuracy while maintaining accuracy on retain classes. The results in Table 1 show: (1) All GU baselines compromise retain-data adversarial robustness (RA_r). (2) Retain-based GU baselines degrade robustness more than retain-free ones. (3) Muter performs unstably for DNNs, regarding both unlearning and robustness. (4) GUARD significantly improves robustness across baselines (at most 6.3× on FT,CIFAR10,ResNet-18), with RA_r close or even superior to Retrain oracle. (5) GUARD also improves clean-data unlearning performance, showing better utility (A_r and A_t) and narrower gap to Retrain oracle. (6) GUARD exhibits consistent superiority in overall performance (Avg.Gap) across datasets and models of varying complexity.

7.3 Random Unlearning

We further evaluate random unlearning by sampling 10% of the training data as $\mathcal{D}_f$, so that $\mathcal{D}_f$ and $\mathcal{D}_r$ share the same distribution. In Table 2, while GU baselines degrade robustness, GUARD effectively preserves it near Retrain levels. Notably, this degradation is milder than in class unlearning, implying that random unlearning requires less drastic modifications to the robust boundaries.

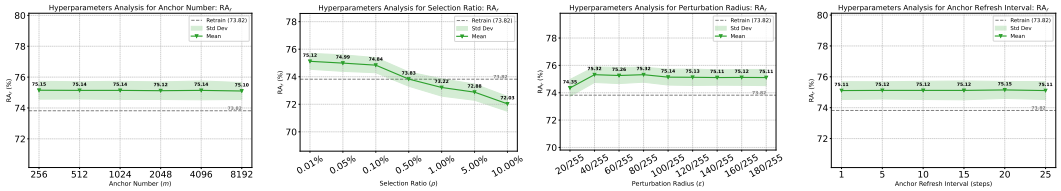
7.4 Ablation Study

Table 3 validates the contributions of *Proj* (Robustness-Aware Gradient Projection) and *Sel* (Importance-Guided Parameter Selection). While applying either module alone improves robustness



(a) FGSM (b) ℓ_∞-PGD (c) CW (d) AutoAttack

Figure 3: RA_r under varying attacks on CIFAR-100 with ResNet-18, using GA as baseline.



(a) Anchor Number (b) Selection Ratio ρ (c) Perturb Radius ϵ (d) Refresh Interval

Figure 4: Hyperparameter analysis using GA with AllCNN on CIFAR-10

and utility, their combination yields optimal complementary effects (e.g., $16\times$ reduction in Avg. Gap on ResNet-18). This consistent synergy across architectures demonstrates strong generalizability.

7.5 Results under Varying Attacks

We evaluate GUARD under 4 adversarial attacks: FGSM, ℓ_∞-PGD-20 [37], CW [5] and AutoAttack [12]. Since we keep the original adversarial training mode and GUARD can only preserve but not introduce additional robustness, all results show large drop under stronger attacks. As shown in Fig. 3, GUARD outperforms the GA baseline across 4 attacks with varying strengths, achieving robustness close to or even better than the Retrain oracle. It confirms that GUARD does not 'overfit' to specific attacks but have broad scalability. See Appendix D.11 for detailed settings and results.

7.6 Hyperparameter Analysis

We analyze GUARD's sensitivity to hyperparameters: anchor number, selection ratio (ρ), perturbation radius for simulating $\mathcal{D}_v$ (ϵ), and anchor refresh interval. Specifically, RA_r maintains stably high in Figure 4a, 4c, 4d. This allows us to reduce computational and storage costs without degrading performance by choosing low anchor number and refresh frequency. Meanwhile, RA_r improves as ρ decreases in Figure 4b, confirming the benefits of Importance-Guided Parameter Selection.

Table 4 shows that increasing the anchor set size improves forgetting with a minor drop in RA. Table 5 demonstrates GUARD's insensitivity to the JL dimension k, where even $k = 64$ achieves performance comparable to exact computation (RA drop $< 1\%$). These results highlight GUARD's scalability and efficiency, as appropriate hyperparameters substantially reduce computational and memory costs with negligible performance trade-offs. Additional results are provided in Appendix D.10.

7.7 Validity of Constructed Adversarially Vulnerable Set

To validate whether CE loss on a small D_v reliably proxies adversarial risk on the full D_r, we track their changes during unlearning and measure Pearson r, Spearman ρ, and regression R^2 across equally sized proxy datasets. As shown in Table 6, both D_f-simulated and D_r-constructed D_v exhibit near-one correlations, confirming their effectiveness as adversarial risk proxies. In contrast, clean D_f, augmented D_f, OOD data (using SVHN dataset [39]), and noise show negative correlations, while clean D_r exhibits slightly weaker rank correlation. These results highlight the effectiveness of our vulnerability set construction.

Table 4: Ablation over anchor number on AllCNN, CIFAR10, GA. Default is 32.

Size	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$RA_r \uparrow$	Update Time (s)	Peak Mem.(GB)
16	0.84	92.67	81.90	75.03	2.55	0.38
32	0.76	92.60	81.91	75.12	3.28	0.43
256	0.02	92.37	81.54	74.76	15.74	1.85
2048	0.00	91.88	81.12	74.45	83.04	11.74
5000	0.00	91.95	81.17	74.98	198.58	23.49

Table 5: Ablation over JL dimension k on AllCNN, CIFAR10, GA. Default is 512.

JL k	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$RA_r \uparrow$	Update Time (s)	Peak Mem.(GB)
64	0.34	92.36	81.58	74.97	2.91	0.19
512	0.76	92.60	81.91	75.12	3.28	0.43
4096	0.82	92.61	81.83	75.20	6.20	2.68
16384	0.82	92.61	81.86	75.20	16.72	10.36
exact	0.82	92.66	81.84	75.44	32.20	0.19

Table 6: Loss-correlation with the adversarial risk on the full D_r on AllCNN, CIFAR-10, GA.

D_v	Pearson $r \uparrow$	Spearman $\rho \uparrow$	$R^2 \uparrow$
D_f-simulated D_v	0.9964	1.0000	0.9928
D_r-constructed D_v	1.0000	1.0000	0.9999
Clean D_r subset	0.9956	0.9762	0.9913
Clean D_f	-0.8986	-0.9762	0.8074
Augmented D_f	-0.9162	-0.9762	0.8393
OOD (SVHN)	-0.7293	-0.5000	0.5319
Noise	-0.8840	-0.9762	0.7815

Table 7: Discrepancies from D_r-constructed D_v on AllCNN, CIFAR-10, GA.

Metric	D_f-simulated D_v	Clean D_f
Accuracy Gap	0.109	0.847
Confidence Gap	0.129	0.673
Margin Gap	0.996	6.803
CE Gap	0.292	4.666
Feature L_2	0.658	1.147
MMD^2	0.108	0.219

Table 8: Time usage (seconds) with AllCNN on CIFAR-10. † denotes GUARD-enhanced variants.

Method	Retrain	Muter	GA	GA†	BS	BS†	SalUn	SalUn†	FT	FT†	UNSIR	UNSIR†
Time (s)	302.4	323.8	3.1	26.4	3.5	26.6	3.9	30.3	10.9	53.2	5.9	30.9

Our D_f-simulated D_v need not semantically match or fully cover the robustness-critical regions of D_r, but should capture their model-level behavior. To validate this, Table 7 evaluates its functional similarity in the model's decision space and proximity in feature representations relative to the D_v constructed from the full D_r, using clean D_f as a baseline. All metrics are computed using randomly selected target retain labels. The simulated D_v exhibits substantially smaller gaps, showing its effectiveness in approximating robustness-critical regions and serving as an informative proxy.

7.8 Efficiency

As shown in Table 8, Muter incurs prohibitive cost due to Hessian computations, with the running time close to the Retrain oracle, limiting its scalability to DNNs. Conversely, GUARD only introduces minor overhead to baselines, maintaining the same order of magnitude. This confirms the efficiency of GUARD in practice. Appendix C.4 details the computational and space complexity analysis.

8 Conclusion

This work proposes GUARD, a scalable and efficient framework to address the adversarial robustness degradation in machine unlearning (MU). In both retain-free and retain-based scenarios, GUARD introduces an adversarial vulnerable dataset construction module, along with two robustness defense modules: Robustness-Aware Gradient Projection and Importance-Guided Parameter Selection. Theoretical and empirical analysis show its superiority over baselines. We expect this work to prompt further research and solutions on MU's side effects. Future directions include defending against adaptive or MU-specific adversarial attacks, and exploring data construction strategies such as GANs.

9 Acknowledgments and Disclosure of Funding

This work is supported by National Natural Science Foundation of China (Grant Nos. U25B2014, 62371069).

References

[1] Bai, L., Tabia, H., Santos-Rodríguez, R.: Adaptive pruning with module robustness sensitivity: Balancing compression and robustness. arXiv preprint arXiv:2410.15176 (2024)

[2] Bartlett, P.L., Mendelson, S.: Rademacher and gaussian complexities: Risk bounds and structural results. Journal of machine learning research **3**(Nov), 463–482 (2002)

[3] Bonato, J., Cotogni, M., Sabetta, L.: Is retain set all you need in machine unlearning? restoring performance of unlearned models with out-of-distribution images. In: European Conference on Computer Vision. pp. 1–19. Springer (2025)

[4] Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C.A., Jia, H., Travers, A., Zhang, B., Lie, D., Papernot, N.: Machine unlearning. In: 2021 IEEE Symposium on Security and Privacy (SP). pp. 141–159. IEEE (2021)

[5] Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. Ieee (2017)

[6] Cha, S., Cho, S., Hwang, D., Lee, H., Moon, T., Lee, M.: Learning to Unlearn: Instance-Wise Unlearning for Pre-trained Classifiers. Proceedings of the AAAI Conference on Artificial Intelligence **38**(10), 11186–11194 (Mar 2024). https://doi.org/10.1609/aaai.v38i10.28996

[7] Chen, H., Zhu, T., Yu, X., Zhou, W.: Machine unlearning via null space calibration. arXiv preprint arXiv:2404.13588 (2024)

[8] Chen, M., Gao, W., Liu, G., Peng, K., Wang, C.: Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7766–7775 (2023)

[9] Chourasia, R., Shah, N.: Forget Unlearning: Towards True Data-Deletion in Machine Learning. In: Proceedings of the 40th International Conference on Machine Learning. pp. 6028–6073. PMLR (Jul 2023)

[10] Chundawat, V.S., Tarun, A.K., Mandal, M., Kankanhalli, M.: Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 7210–7217 (2023)

[11] Cotogni, M., Bonato, J., Sabetta, L., Pelosin, F., Nicolosi, A.: DUCK: Distance-based Unlearning via Centroid Kinematics. https://arxiv.org/abs/2312.02052v2 (Dec 2023)

[12] Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: International conference on machine learning. pp. 2206–2216. PMLR (2020)

[13] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)

[14] Dosovitskiy, A.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

[15] Fan, C., Liu, J., Zhang, Y., Wong, E., Wei, D., Liu, S.: Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. arXiv preprint arXiv:2310.12508 (2023)

[16] Foster, J., Schoepf, S., Brintrup, A.: Fast Machine Unlearning without Retraining through Selective Synaptic Dampening. Proceedings of the AAAI Conference on Artificial Intelligence **38**(11), 12043–12051 (Mar 2024). https://doi.org/10.1609/aaai.v38i11.29092

[17] Golatkar, A., Achille, A., Soatto, S.: Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9301–9309. IEEE, Seattle, WA, USA (Jun 2020). https://doi.org/10.1109/CVPR42600.2020.00932

[18] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)

[19] Guo, C., Goldstein, T., Hannun, A., Van Der Maaten, L.: Certified data removal from machine learning models. In: Proceedings of the 37th International Conference on Machine Learning. ICML'20, JMLR.org (2020)

[20] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[21] Hoang, T., Rana, S., Gupta, S., Venkatesh, S.: Learn to unlearn for deep neural networks: Minimizing unlearning interference with gradient projection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4819–4828 (2024)

[22] Hua, A., Gu, J., Xue, Z., Carlini, N., Wong, E., Qin, Y.: Initialization matters for adversarial transfer learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24831–24840 (2024)

[23] Huang, Y., Liu, D., Chua, L., Ghazi, B., Kamath, P., Kumar, R., Manurangsi, P., Nasr, M., Sinha, A., Zhang, C.: Unlearn and burn: Adversarial machine unlearning requests destroy model accuracy. arXiv preprint arXiv:2410.09591 (2024)

[24] Huang, Z., Cheng, X., Zheng, J., Wang, H., He, Z., Li, T., Huang, X.: Unified gradient-based machine unlearning with remain geometry enhancement. Advances in Neural Information Processing Systems **37**, 26377–26414 (2024)

[25] Jordao, A., Pedrini, H.: On the effect of pruning on adversarial robustness. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1–11 (2021)

[26] Jung, Y., Cho, I., Hsu, S.H., Hockenmaier, J.: Attack and reset for unlearning: Exploiting adversarial noise toward machine unlearning through parameter re-initialization. arXiv preprint arXiv:2401.08998 (2024)

[27] Kim, H., Lee, S., Woo, S.S.: Layer attack unlearning: Fast and accurate machine unlearning via layer level attack and knowledge distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 21241–21248 (2024)

[28] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

[29] Kurmanji, M., Triantafillou, P., Hayes, J., Triantafillou, E.: Towards unbounded machine unlearning. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2024)

[30] Li, G., Duggal, R., Singh, A., Kundu, K., Shuai, B., Wu, J.: Robustness preserving fine-tuning using neuron importance. In: European Conference on Computer Vision. pp. 54–69. Springer (2024)

[31] Li, K., Ferrand, J.C.N., Sheatsley, R., Hoak, B., Beugin, Y., Pauley, E., McDaniel, P.: On the robustness tradeoff in fine-tuning. arXiv preprint arXiv:2503.14836 (2025)

[32] Lin, S., Zhang, X., Chen, C., Chen, X., Susilo, W.: ERM-KTP: Knowledge-Level Machine Unlearning via Knowledge Transfer. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20147–20155. IEEE, Vancouver, BC, Canada (Jun 2023). https://doi.org/10.1109/CVPR52729.2023.01929

[33] Liu, J., Ram, P., Yao, Y., Liu, G., Liu, Y., SHARMA, P., Liu, S., et al.: Model sparsity can simplify machine unlearning. Advances in Neural Information Processing Systems **36** (2024)

[34] Liu, J., Lou, J., Qin, Z., Ren, K.: Certified minimax unlearning with generalization rates and deletion capacity. Advances in Neural Information Processing Systems **36**, 62821–62852 (2023)

[35] Liu, J., Xue, M., Lou, J., Zhang, X., Xiong, L., Qin, Z.: Muter: Machine unlearning on adversarially trained models. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4892–4902 (2023)

[36] Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)

[37] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)

[38] Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. MIT Press, second edn. (2018)

[39] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y., et al.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning. vol. 2011, p. 4. Granada (2011)

[40] Panda, S., Sourav, S., A.P., P.: Partially blinded unlearning: Class unlearning for deep networks from bayesian perspective. Proceedings of the AAAI Conference on Artificial Intelligence **39**(6), 6372–6380 (Apr 2025). https://doi.org/10.1609/aaai.v39i6.32682, https://ojs.aaai.org/index.php/AAAI/article/view/32682

[41] Sehwag, V., Wang, S., Mittal, P., Jana, S.: Hydra: Pruning adversarially robust neural networks. Advances in Neural Information Processing Systems **33**, 19655–19666 (2020)

[42] Shafahi, A., Saadatpanah, P., Zhu, C., Ghiasi, A., Studer, C., Jacobs, D., Goldstein, T.: Adversarially robust transfer learning. arXiv preprint arXiv:1905.08232 (2019)

[43] Shamsian, A., Shaar, E., Navon, A., Chechik, G., Fetaya, E.: Go beyond your means: Unlearning with per-sample gradient orthogonalization. arXiv preprint arXiv:2503.02312 (2025)

[44] Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)

[45] Tarun, A.K., Chundawat, V.S., Mandal, M., Kankanhalli, M.: Fast yet effective machine unlearning. IEEE Transactions on Neural Networks and Learning Systems **35**(9), 13046–13055 (2023)

[46] Voigt, P., Von dem Bussche, A.: The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing **10**(3152676), 10–5555 (2017)

[47] Warnecke, A., Pirch, L., Wressnegger, C., Rieck, K.: Machine unlearning of features and labels. arXiv preprint arXiv:2108.11577 (2021)

[48] Wei, S., Zhang, M., Zha, H., Wu, B.: Shared adversarial unlearning: Backdoor mitigation by unlearning shared adversarial examples. Advances in Neural Information Processing Systems **36**, 25876–25909 (2023)

[49] Xu, H., Zhu, T., Zhang, L., Zhou, W., Yu, P.S.: Machine unlearning: A survey. ACM Comput. Surv. **56**(1) (Aug 2023). https://doi.org/10.1145/3603620, https://doi.org/10.1145/3603620

[50] Zhang, C., Liu, A., Liu, X., Xu, Y., Yu, H., Ma, Y., Li, T.: Interpreting and improving adversarial robustness of deep neural networks with neuron sensitivity. IEEE Transactions on Image Processing **30**, 1291–1304 (2020)

[51] Zhao, C., Qian, W., Li, Y., Chen, A., Huai, M.: Rethinking adversarial robustness in the context of the right to be forgotten. In: Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., Berkenkamp, F. (eds.) Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 60927–60939. PMLR (21–27 Jul 2024), https://proceedings.mlr.press/v235/zhao24k.html

[52] Zhu, K., Hu, X., Wang, J., Xie, X., Yang, G.: Improving generalization of adversarial training via robust critical fine-tuning. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4424–4434 (2023)

Appendix

A Pseudo Codes

Algorithm 1 Construction of Adversarially Vulnerable Dataset $\mathcal{D}_v$

Input: Retain set $\mathcal{D}_r$ (optional), Forget set $\mathcal{D}_f$, Model parameters θ, Sampling ratio γ, Perturbation budget ε, Iterations T, Augmentation set $\mathcal{T}$.
Output: Adversarially Vulnerable Set $\mathcal{D}_v$.
Initialize $\mathcal{D}_v \leftarrow \emptyset$.
if $\mathcal{D}_r$ is accessible **then**
  Sample a subset $\mathcal{D}_r^{\text{sub}} \subset \mathcal{D}_r$ with size $\gamma|\mathcal{D}_r|$.
  for each $(x, y) \in \mathcal{D}_r^{\text{sub}}$ **do**
    Generate perturbation: $\tilde{x} \leftarrow \text{PGD}_{\varepsilon_1,p}^{(T)}(x, y; \theta)$.
    if $f_\theta(\tilde{x}) = y_r$ **then**
      $\mathcal{D}_v \leftarrow \mathcal{D}_v \cup \{(\tilde{x}, y)\}$.
    end if
  end for
else
  {Simulate $\mathcal{D}_v$ using Forget Data}
  for each $(x^f, y^f) \in \mathcal{D}_f$ **do**
    Apply semantic augmentation: $\hat{x} \leftarrow \mathcal{T}(x^f)$.
    Assign random target retain label y^r.
    Generate targeted perturbation: $\tilde{x} \leftarrow \text{PGD}_{\varepsilon_2,p}^{(T)}(\hat{x}, y^r; \theta)$.
    if $f_\theta(\tilde{x}) = y^r$ **then**
      $\mathcal{D}_v \leftarrow \mathcal{D}_v \cup \{(\tilde{x}, y^r)\}$.
    end if
  end for
end if
return $\mathcal{D}_v$

Algorithm 2 Robustness-Aware Gradient Projection

Input: Model parameters θ, Original update direction $p = -P\nabla_\theta \mathcal{L}_{\text{unl}}$, Adversarially vulnerable set $\mathcal{D}_v = \{(x_i^v, y_i^v)\}_{i=1}^m$, Preconditioner P, Tolerance κ.
Output: Projected update direction q.
Compute Jacobian $J \in \mathbb{R}^{m \times d}$ where row $J_i^\top = \nabla_\theta \ell_{\text{CE}}(f_\theta(x_i^v), y_i^v)$.
{Check for constraint violations}
Calculate violation vector: $v \leftarrow Jp - \kappa$.
if $\max(v) \leq 0$ **then**
  $q \leftarrow p$
else
  Compute Gram matrix $S \leftarrow JPJ^\top$.
  Compute active set violation: $b \leftarrow [v]_+$.
  Solve linear system for α: $S\alpha = b$.
  $q \leftarrow p - PJ^\top \alpha$.
end if
return q

B Detailed Derivation for Robustness-Aware Gradient Projection

This section provides the explicit derivation for the solution to the constrained optimization problem posed in the main text Section 5.2.

Algorithm 3 Importance-Guided Parameter Selection

Input: Model parameters $\theta \in \mathbb{R}^d$, Forget set $\mathcal{D}_f$, Adversarially vulnerable set $\mathcal{D}_v$, Percentile threshold ρ, Small constant ϵ.
Output: Binary Mask $M \in \{0,1\}^d$.
Calculate forgetting importance for all d parameters:
$I^{(\mathrm{f})} \leftarrow \frac{1}{|\mathcal{D}_f|} \sum_{(x,y)\in\mathcal{D}_f} |\nabla_\theta \ell_{\mathrm{CE}}(f_\theta(x), y)|$.
Calculate adversarial robustness importance for all d parameters:
$I^{(\mathrm{adv})} \leftarrow \frac{1}{|\mathcal{D}_v|} \sum_{(x,y)\in\mathcal{D}_v} |\nabla_\theta \ell_{\mathrm{CE}}(f_\theta(x), y)|$.
Compute ratio scores for all d parameters:
$S \leftarrow I^{(\mathrm{f})}/(I^{(\mathrm{adv})} + \epsilon)$.
Generate Binary Mask, seting mask entries to 1 for parameters with top ρ-percentile scores:
$M_k \leftarrow \mathbb{I}\left[S_k \in \mathrm{TopK}(S, \rho)\right], \quad \forall k \in \{1, \ldots, d\}$.
return M

B.1 Lagrangian Derivation

Recall the optimization objective is to minimize the deviation from the original update p under the P^{-1}-norm, subject to linear constraints $Jq \le \kappa$. The primal problem is:

$$\min_{q\in\mathbb{R}^d} \quad \frac{1}{2}\|q-p\|^2_{P^{-1}} = \frac{1}{2}(q-p)^\top P^{-1}(q-p) \qquad \text{s.t.} \quad Jq - \kappa \le 0. \tag{11}$$

We introduce the dual variables (Lagrange multipliers) $\alpha \in \mathbb{R}^m, \alpha \ge 0$. The Lagrangian function $\mathcal{L}(q, \alpha)$ is defined as:

$$\mathcal{L}(q,\alpha) = \frac{1}{2}(q-p)^\top P^{-1}(q-p) + \alpha^\top (Jq - \kappa). \tag{12}$$

The Karush-Kuhn-Tucker (KKT) stationarity condition requires the gradient of $\mathcal{L}$ with respect to q to vanish at the optimum q^*:

$$\nabla_q \mathcal{L}(q^*, \alpha) = P^{-1}(q^* - p) + J^\top \alpha = 0. \tag{13}$$

Since the preconditioner P is positive definite, multiplying both sides by P yields the expression for the updated direction:

$$q^* = p - PJ^\top \alpha. \tag{14}$$

This equation confirms that the correction term $-PJ^\top\alpha$ resides in the subspace spanned by the preconditioned gradients of the robustness anchors.

B.2 Resolution of Dual Variables

To determine α, we consider the complementary slackness condition $\alpha_i(J_i q^* - \kappa_i) = 0$. For the active constraints where $J_i q^* = \kappa_i$ (i.e., the robustness boundary is hit), we substitute Eq. (14) into the equality $Jq^* = \kappa$:

$$\begin{aligned} J(p - PJ^\top\alpha) &= \kappa \\ Jp - (JPJ^\top)\alpha &= \kappa \\ (JPJ^\top)\alpha &= Jp - \kappa. \end{aligned} \tag{15}$$

Let $G = JPJ^\top \in \mathbb{R}^{m\times m}$ be the Gram matrix of the anchor gradients scaled by P. Assuming G is invertible (in practice, we add a damping term to satisfy this assumption), we have $\alpha = G^{-1}(Jp - \kappa)$.

Since the original constraints are inequalities ($Jq \le \kappa$), the multiplier α_i must be non-negative and non-zero only when the constraint is violated or active. We enforce this by projecting the violation term $Jp - \kappa$ onto the non-negative orthant. This yields the closed-form approximation for the multipliers:

$$\alpha = (JPJ^\top)^{-1}[Jp - \kappa]_+, \tag{16}$$

where $[\cdot]_+$ applies $\max(0, \cdot)$ element-wise. Substituting this back into Eq. (14) recovers the final update rule presented in the main text:

$$q = p - PJ^\top (JPJ^\top)^{-1}[Jp - \kappa]_+. \tag{17}$$

B.3 Geometric Consistency with Preconditioned Optimization

The derived update rule $q = p - PJ^\top (JPJ^\top)^{-1}[Jp - \kappa]_+$ represents a projection of the original update p onto the robustness-constrained feasible region. A critical feature of our design is that this projection minimizes the deviation under the P^{-1}-norm, i.e., $\|q - p\|^2_{P^{-1}}$, rather than the standard Euclidean norm $\|q - p\|^2_2$.

This distinction is fundamental for compatibility with adaptive optimization algorithms. Standard Euclidean projection assumes an isotropic parameter space where a unit change in any parameter carries equal weight. However, adaptive optimizers (e.g., Adam) implicitly operate on a Riemannian manifold by using a preconditioner P (approximating the inverse Hessian) to rescale gradients according to local curvature. This anisotropic scaling accelerates convergence in flat directions while damping oscillations in steep ones. Applying a standard Euclidean projection in this context would ignore the learned geometry, potentially penalizing the accelerated updates in flat directions as "large deviations" and effectively undoing the optimizer's preconditioning benefits.

In contrast, our formulation ensures *metric consistency*. By incorporating P into the projection operator, we measure the "closeness" of q to p using the same metric that generated p. Geometrically, the preconditioner P transforms the elliptical cost landscape into a more spherical geometry; our operator performs an orthogonal projection within this transformed space. This ensures that the correction term $-PJ^\top \alpha$ is orthogonal to the safe subspace with respect to the local curvature, modifying the update vector only along the directions that genuinely violate the robustness constraints without disrupting the relative scaling of parameters established by the optimizer.

B.4 Johnson–Lindenstrauss (JL) Approximation

In this section, we provide the complexity analysis that necessitates dimensionality reduction and the theoretical derivation connecting the Johnson–Lindenstrauss (JL) Lemma to the inner product preservation required by our method.

B.4.1 Computational and Storage Bottlenecks in Gradient Projection

The core of our method involves solving the constrained quadratic programming (QP) problem to find the dual variables α:

$$S\alpha = [b]_+, \quad \text{with } S = JPJ^\top \text{ and } b = Jp - \kappa. \tag{18}$$

Here, $J \in \mathbb{R}^{m \times d}$ denotes the Jacobian. The computational cost of this step decomposes into two phases: **construction** (forming S and b) and **solution** (solving for α).

The **solution** phase typically scales with m (e.g., $O(m^3)$ via Cholesky decomposition). Since the number of constraints m is usually small relative to the parameter dimension d ($m \ll d$), this cost is negligible. However, the **construction** phase presents the true bottleneck as it depends linearly on the high dimension d. Specifically, constructing the Gram matrix S exactly requires computing pairwise inner products, costing $O(m^2 d)$. As m increases to improve robustness, this term dominates the computation. Furthermore, explicitly storing the Jacobian J incurs an $O(md)$ memory overhead. For large-scale models, this results in a prohibitive storage requirement (often gigabytes of gradient cache) even for a small number of constraints. JL approximation is used to address these limitations by decoupling both the computational and storage costs from the parameter dimension d.

B.4.2 Detailed Derivation of JL Approximation

Proposition B.1. *Let Φ be a random projection matrix that satisfies the Johnson-Lindenstrauss (JL) property with tolerance $\varepsilon \in (0, 1)$. Defining $v_i = P^{1/2}\nabla_\theta r_i(\theta)$, $u = P^{-1/2}p$, so that $S_{ij} = v_i^\top v_j$ and $(Jp)_i = v_i^\top u$. Let $\tilde{v}_i = \Phi v_i$ and $\tilde{u} = \Phi u$. Then, the inner product in the projected space approximates the original inner product:*

$$S_{ij} \approx \tilde{v}_i^\top \tilde{v}_j, \tag{19}$$

$$(Jp)_i \approx \tilde{v}_i^\top \tilde{u}. \tag{20}$$

Proof. The standard Johnson-Lindenstrauss Lemma guarantees the preservation of Euclidean distances. Specifically, for any pair of vectors x, y in the set of interest, the projection Φ satisfies the

following inequality with high probability:

$$(1-\beta)\|x-y\|^2 \leq \|\Phi x - \Phi y\|^2 \leq (1+\beta)\|x-y\|^2. \tag{21}$$

While the JL Lemma directly bounds the distortion of squared norms, our goal is to approximate the Gram matrix entries, which are inner products. To bridge this gap, we invoke the **Polarization Identity**, which expresses an inner product entirely in terms of norms:

$$x^\top y = \frac{1}{2}\left(\|x\|^2 + \|y\|^2 - \|x-y\|^2\right). \tag{22}$$

Since Φ is a linear map that approximately preserves squared norms (i.e., distances to the origin, taking $y = 0$ in the JL lemma) and squared pairwise Euclidean distances, it consequently preserves the inner product structure.

Substituting the definition of the embedding vectors $\tilde{v} = \Phi v$ into the polarization identity, we derive:

$$\begin{aligned}
\tilde{v}_i^\top \tilde{v}_j &= \frac{1}{2}\left(\|\Phi v_i\|^2 + \|\Phi v_j\|^2 - \|\Phi v_i - \Phi v_j\|^2\right) \\
&\approx \frac{1}{2}\left(\|v_i\|^2 + \|v_j\|^2 - \|v_i - v_j\|^2\right) \\
&= v_i^\top v_j = S_{ij}.
\end{aligned} \tag{23}$$

Similarly, for the interaction between a basis vector v_i and an arbitrary vector u (mapped to $\tilde{u} = \Phi u$), we have:

$$\begin{aligned}
\tilde{v}_i^\top \tilde{u} &= \frac{1}{2}\left(\|\Phi v_i\|^2 + \|\Phi u\|^2 - \|\Phi v_i - \Phi u\|^2\right) \\
&\approx \frac{1}{2}\left(\|v_i\|^2 + \|u\|^2 - \|v_i - u\|^2\right) \\
&= v_i^\top u = (Jp)_i.
\end{aligned} \tag{24}$$

This confirms that minimizing geometric distortion in the projected space (via the JL Lemma) is mathematically equivalent to maintaining the spectral properties of the linear system defined by the Gram matrix. □

B.4.3 Streaming JL Projection

The main text mentions using a streaming JL projection to save memory. This is essential because a standard implementation requires storing the dense random matrix $\Phi \in \mathbb{R}^{k\times d}$, incurring an $O(kd)$ memory overhead that can be prohibitive for high-dimensional models. Specifically, instead of instantiating Φ, we generate its entries on-the-fly using a pseudo-random number generator (PRNG) with a fixed seed. As the gradient elements are computed during backpropagation, they are immediately aggregated into the low-dimensional sketches $\tilde{v}_i$. This reduces the auxiliary storage requirement from $O(kd)$ (for Φ) to merely $O(k)$ for the PRNG state, making the method memory-efficient and scalable to large-scale training.

C Detailed Theoretical Analysis

C.1 Detailed Proof for Adversarially Vulnerable Set Construction

In this section, we provide the detailed proofs for the uniform convergence of the constructed adversarially vulnerable dataset $\mathcal{D}_v$ and the subsequent Eq.(8). The proof follows the standard symmetrization argument and concentration inequalities found in statistical learning theory literature [38, 2]. We provide the detailed derivation adapted to our notation and specific setting for completeness.

Proof. Let $\Phi(\mathcal{D}_v) = \sup_{\theta\in\Theta} |\mathcal{R}^*(\theta) - \mathcal{R}(\theta)|$ denote the uniform deviation. Since the loss is bounded by M, changing one sample in $\mathcal{D}_v$ changes $\Phi(\mathcal{D}_v)$ by at most M/m. By McDiarmid's inequality, with probability at least $1 - \delta/2$, the deviation concentrates around its expectation:

$$\Phi(\mathcal{D}_v) \leq \mathbb{E}_{\mathcal{D}_v}[\Phi(\mathcal{D}_v)] + M\sqrt{\frac{\log(2/\delta)}{2m}}. \tag{25}$$

Standard symmetrization arguments bound the expectation by twice the expected Rademacher complexity, i.e., $\mathbb{E}[\Phi(\mathcal{D}_v)] \leq 2\mathbb{E}[\mathfrak{R}_m(\mathcal{F})]$. Furthermore, the empirical Rademacher complexity $\mathfrak{R}_m(\mathcal{F})$ itself satisfies the bounded difference property. Applying McDiarmid's inequality again, with probability at least $1 - \delta/2$, we have:

$$\mathbb{E}[\mathfrak{R}_m(\mathcal{F})] \leq \mathfrak{R}_m(\mathcal{F}) + M\sqrt{\frac{\log(2/\delta)}{2m}}. \tag{26}$$

By union bound, both inequalities Eq. (25) and Eq. (26) hold simultaneously with probability at least $1 - \delta$. Substituting Eq. (26) into Eq. (25), we have:

$$\sup_{\theta\in\Theta} |\mathcal{R}^*(\theta) - \mathcal{R}(\theta)| \leq 2\left(\mathfrak{R}_m(\mathcal{F}) + M\sqrt{\frac{\log(2/\delta)}{2m}}\right) + M\sqrt{\frac{\log(2/\delta)}{2m}} \tag{27}$$

$$= 2\mathfrak{R}_m(\mathcal{F}) + 3M\sqrt{\frac{\log(2/\delta)}{2m}}. \tag{28}$$

This completes the proof. □

C.1.1 Proof of Eq.(8)

Proof. When the event in the uniform convergence holds (with probability $1 - \delta$), for any $\theta \in \Theta$, we have the two-sided inequality:

$$\mathcal{R}(\theta) - \epsilon_{\text{gen}} \leq \mathcal{R}^*(\theta) \leq \mathcal{R}(\theta) + \epsilon_{\text{gen}}. \tag{29}$$

Applying the inequality above to both θ_{unl} and θ_{ori}:

$$\mathcal{R}^*(\theta_{\text{unl}}) \leq \mathcal{R}(\theta_{\text{unl}}) + \epsilon_{\text{gen}}, \tag{30}$$

$$\mathcal{R}^*(\theta_{\text{ori}}) \geq \mathcal{R}(\theta_{\text{ori}}) - \epsilon_{\text{gen}}. \tag{31}$$

Subtracting the second inequality from the first:

$$\mathcal{R}^*(\theta_{\text{unl}}) - \mathcal{R}^*(\theta_{\text{ori}}) \leq (\mathcal{R}(\theta_{\text{unl}}) + \epsilon_{\text{gen}}) - (\mathcal{R}(\theta_{\text{ori}}) - \epsilon_{\text{gen}}) \tag{32}$$

$$= (\mathcal{R}(\theta_{\text{unl}}) - \mathcal{R}(\theta_{\text{ori}})) + 2\epsilon_{\text{gen}}. \tag{33}$$

Substituting the definitions $\Delta\mathcal{R}^* = \mathcal{R}^*(\theta_{\text{unl}}) - \mathcal{R}^*(\theta_{\text{ori}})$ and $\Delta\mathcal{R} = \mathcal{R}(\theta_{\text{unl}}) - \mathcal{R}(\theta_{\text{ori}})$, we obtain:

$$\Delta\mathcal{R}^* \leq \Delta\mathcal{R} + 2\epsilon_{\text{gen}}. \tag{34}$$

This concludes the proof. □

C.2 Detailed Proof for Robustness-Aware Gradient Projection

In this section, we provide the complete proofs for the theorems presented in Section 6.2. We first formally define the optimization problem and an auxiliary lemma regarding the boundedness of the update direction, followed by the proofs of Theorem 6.2 and Proposition 6.3.

C.2.1 Preliminaries and Auxiliary Lemma

Recall that at each time step t, the standard gradient update direction (without projection) is given by $p_t = -P\nabla\mathcal{L}_{\text{unl}}(\theta_t)$, where P is the positive definite preconditioner. The actual parameter update corresponds to $\theta_{t+1} = \theta_t + \eta q_t$, where η is the step size. Our Robustness-Aware Gradient Projection strategy seeks an update direction q_t that minimizes the deviation from p_t while satisfying the robustness constraints:

$$q_t = \arg\min_{q\in\mathbb{R}^d} \frac{1}{2}\|q - p_t\|^2_{P^{-1}} \quad \text{s.t.} \quad \nabla r_i(\theta_t)^\top q \leq \kappa, \quad \forall i \in \{1, \ldots, m\}, \tag{35}$$

where $\nabla r_i(\theta_t)$ is the gradient of the robustness indicator for the i-th anchor. Let $\mathcal{C}_t = \{q \in \mathbb{R}^d : \nabla r_i(\theta_t)^\top q \leq \kappa, \forall i\}$ denote the feasible set for q.

Before proving the main theorem, we establish an upper bound on the norm of the projected update direction q_t.

Lemma C.1 (Boundedness of Update Directions)**.** *Under Assumption 6.1, let $B = \lambda_{\max}(P)G$. The Euclidean norms of both the unconstrained update p_t and the projected update q_t are bounded by the same constant B:*

$$\|q_t\|_2 \leq B \quad \textit{and} \quad \|p_t\|_2 \leq B. \tag{36}$$

Proof. We first derive the bound for the unconstrained update $p_t = -P\nabla\mathcal{L}_{\text{unl}}(\theta_t)$. Using the definition of the spectral norm and Assumption 6.1:

$$\|p_t\|_2 = \| - P\nabla\mathcal{L}_{\text{unl}}(\theta_t)\|_2 \leq \|P\|_2\|\nabla\mathcal{L}_{\text{unl}}(\theta_t)\|_2 \leq \lambda_{\max}(P)G = B. \tag{37}$$

Next, we analyze the projected update q_t. Recall that q_t is the projection of p_t onto $\mathcal{C}_t$ under the P^{-1}-norm. Since the origin is feasible ($0 \in \mathcal{C}_t$), the standard non-expansive property of metric projection implies that the norm does not increase in the projection metric:

$$\|q_t\|_{P^{-1}} \leq \|p_t\|_{P^{-1}}. \tag{38}$$

We now derive the bound for q_t in the Euclidean norm. First, we bound $\|p_t\|_{P^{-1}}$ using the Rayleigh quotient property:

$$\|p_t\|_{P^{-1}}^2 = \nabla\mathcal{L}_{\text{unl}}(\theta_t)^\top P\nabla\mathcal{L}_{\text{unl}}(\theta_t) \leq \lambda_{\max}(P)\|\nabla\mathcal{L}_{\text{unl}}(\theta_t)\|_2^2 \leq \lambda_{\max}(P)G^2. \tag{39}$$

Thus, $\|p_t\|_{P^{-1}} \leq \sqrt{\lambda_{\max}(P)}G$. Substituting this into Eq. (38), we have $\|q_t\|_{P^{-1}} \leq \sqrt{\lambda_{\max}(P)}G$. Finally, converting the P^{-1}-norm back to the Euclidean norm (using $\|v\|_2 \leq \sqrt{\lambda_{\max}(P)}\|v\|_{P^{-1}}$):

$$\|q_t\|_2 \leq \sqrt{\lambda_{\max}(P)}\|q_t\|_{P^{-1}} \leq \sqrt{\lambda_{\max}(P)} \cdot \sqrt{\lambda_{\max}(P)}G = \lambda_{\max}(P)G = B. \tag{40}$$

This concludes that both $\|p_t\|_2$ and $\|q_t\|_2$ are bounded by B. □

C.2.2 Proof of Theorem 6.2

Proof. We aim to bound the total increment in the empirical robustness indicator $\Delta R_{\text{proj}} = \mathcal{R}(\theta_T) - \mathcal{R}(\theta_0)$, where $\theta_0 = \theta_{\text{ori}}$ and $\theta_T = \theta_{\text{unl}}^{\text{proj}}$. The total change can be decomposed into the sum of changes at each step:

$$\Delta R_{\text{proj}} = \sum_{t=0}^{T-1} \left(\mathcal{R}(\theta_{t+1}) - \mathcal{R}(\theta_t)\right). \tag{41}$$

Recall that $\mathcal{R}(\theta) = \frac{1}{m}\sum_{i=1}^m r_i(\theta)$. We first analyze the change in the individual robustness indicator $r_i(\theta)$ for a single step update $\theta_{t+1} = \theta_t + \eta q_t$.

By Assumption 6.1, $r_i(\theta)$ is L_i-smooth. The quadratic upper bound for smooth functions gives:

$$r_i(\theta_{t+1}) - r_i(\theta_t) \leq \eta\nabla r_i(\theta_t)^\top q_t + \frac{L_i}{2}\eta^2\|q_t\|_2^2. \tag{42}$$

By the construction of our Robustness-Aware Gradient Projection (Eq. (35)), the update direction q_t must satisfy the linear constraint $\nabla r_i(\theta_t)^\top q_t \leq \kappa$ for all anchors i. Furthermore, using Lemma C.1, we have $\|q_t\|_2 \leq B$. Substituting these into the inequality:

$$r_i(\theta_{t+1}) - r_i(\theta_t) \leq \eta\kappa + \frac{L_i}{2}\eta^2 B^2. \tag{43}$$

Now, we aggregate this bound over all anchors to find the bound for the empirical risk $\mathcal{R}(\theta)$:

$$\mathcal{R}(\theta_{t+1}) - \mathcal{R}(\theta_t) = \frac{1}{m}\sum_{i=1}^m \left(r_i(\theta_{t+1}) - r_i(\theta_t)\right) \tag{44}$$

$$\leq \frac{1}{m}\sum_{i=1}^m \left(\eta\kappa + \frac{L_i}{2}\eta^2 B^2\right) \tag{45}$$

$$= \eta\kappa + \frac{\eta^2 B^2}{2}\left(\frac{1}{m}\sum_{i=1}^m L_i\right). \tag{46}$$

Let $\bar{L} = \max_i L_i$. It holds that $\frac{1}{m}\sum_{i=1}^{m} L_i \leq \bar{L}$. Thus, the single-step degradation is bounded by:

$$\mathcal{R}(\theta_{t+1}) - \mathcal{R}(\theta_t) \leq \eta\kappa + \frac{\bar{L}}{2}\eta^2 B^2. \tag{47}$$

Summing over T steps from $t = 0$ to $T - 1$:

$$\Delta R_{\text{proj}} \leq \sum_{t=0}^{T-1} \left(\eta\kappa + \frac{\bar{L}}{2}\eta^2 B^2\right) = T\left(\eta\kappa + \frac{\bar{L}}{2}\eta^2 B^2\right). \tag{48}$$

This completes the proof. □

C.2.3 Proof of Proposition 6.3

Proof. We compare the robustness degradation bound of our method (Theorem 6.2) with that of the baseline standard unlearning.

Baseline Analysis: For the baseline, the parameters are updated using the unconstrained direction $p_t = -P\nabla\mathcal{L}_{\text{unl}}(\theta_t)$, resulting in the update rule $\theta_{t+1} = \theta_t + \eta p_t$. According to Lemma C.1, the norm of the baseline update direction is bounded by the same constant B as the projected update:$\|p_t\|_2 \leq B$.

Applying the smoothness assumption to the baseline update step ηp_t:

$$\mathcal{R}(\theta_{t+1}) - \mathcal{R}(\theta_t) \leq \nabla\mathcal{R}(\theta_t)^\top(\eta p_t) + \frac{\bar{L}}{2}\|\eta p_t\|_2^2 \leq \eta\nabla\mathcal{R}(\theta_t)^\top p_t + \frac{\bar{L}}{2}\eta^2 B^2. \tag{49}$$

Summing over T steps to obtain the total degradation ΔR_{base}:

$$\Delta R_{\text{base}} \leq \sum_{t=0}^{T-1} \left(\eta\nabla\mathcal{R}(\theta_t)^\top p_t + \frac{\bar{L}}{2}\eta^2 B^2\right) \tag{50}$$

$$\leq T \cdot \eta \underbrace{\left(\frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{m}\sum_{i=1}^{m}\nabla r_i(\theta_t)^\top p_t\right)}_{:=\bar{\mathcal{A}}_{\text{base}}} + T \cdot \frac{\bar{L}}{2}\eta^2 B^2. \tag{51}$$

Thus, we define the upper bound for the baseline as:

$$\text{Bound}(\Delta R_{\text{base}}) \coloneqq T\left(\eta\bar{\mathcal{A}}_{\text{base}} + \frac{\bar{L}}{2}\eta^2 B^2\right). \tag{52}$$

Here, $\bar{\mathcal{A}}_{\text{base}}$ represents the average gradient conflict between the unlearning direction and the robustness objectives.

Comparison: Recall the bound for Robustness-Aware Gradient Projection from Theorem 6.2:

$$\text{Bound}(\Delta R_{\text{proj}}) = T\left(\eta\kappa + \frac{\bar{L}}{2}\eta^2 B^2\right). \tag{53}$$

Comparing the two bounds, we observe that the second-order terms ($O(\eta^2)$) are **identical** because both $\|p_t\|_2$ and $\|q_t\|_2$ are bounded by B. Consequently, the comparison reduces strictly to the first-order linear terms: $\eta\kappa$ versus $\eta\bar{\mathcal{A}}_{\text{base}}$.

The term $\bar{\mathcal{A}}_{\text{base}}$ is typically positive and large, as standard unlearning inherently conflicts with robustness (i.e., p_t tends to increase $r_i(\theta)$). By setting the tolerance parameter κ such that:

$$\kappa < \bar{\mathcal{A}}_{\text{base}}, \tag{54}$$

we guarantee strictly tighter control over robustness degradation regardless of the step size η or the bound magnitude B. Since we can arbitrarily choose a small κ (e.g., $\kappa \approx 0$), our method ensures:

$$\text{Bound}(\Delta R_{\text{proj}}) < \text{Bound}(\Delta R_{\text{base}}). \tag{55}$$

This completes the proof. □

C.3 Detailed Proof for Importance-Guided Parameter Selection

In this section, we provide the complete proofs for the theorems presented in Section 6.3.

C.3.1 Preliminaries and Assumptions

As detailed in the method section, we quantify parameter importance using the average gradient magnitude on the respective datasets. Recall the *forgetting importance* $I_k^{(\mathrm{f})}$ on the forget set $\mathcal{D}_f$ and the *robustness importance* $I_k^{(\mathrm{adv})}$ on the adversarial anchor set $\mathcal{D}_v$:

$$I_k^{(\mathrm{f})} = \frac{1}{|\mathcal{D}_f|} \sum_{(x,y)\in\mathcal{D}_f} \left|\nabla_{\theta_k} \mathcal{L}_{\mathrm{unl}}(\theta; x, y)\right|, \tag{56}$$

$$I_k^{(\mathrm{adv})} = \frac{1}{|\mathcal{D}_v|} \sum_{(x,y)\in\mathcal{D}_v} \left|\nabla_{\theta_k} \mathcal{R}(\theta; x, y)\right|. \tag{57}$$

To identify parameters critical for forgetting but insensitive to robustness, we compute the *ratio score* s_k:

$$s_k = \frac{I_k^{(\mathrm{f})}}{I_k^{(\mathrm{adv})} + \epsilon}, \tag{58}$$

where $\epsilon > 0$ is a small constant for numerical stability.

Our method retains the top ρ fraction of parameters with the largest ratio scores. This selection strategy effectively induces a threshold τ_{sel} (specifically, the $(1-\rho)$-quantile of the ratio distribution $\{s_k\}_{k=1}^d$), such that the set of selected parameter indices S satisfies:

$$k \in S \iff s_k \geq \tau_{\mathrm{sel}}. \tag{59}$$

The parameter update is defined as $\theta_{t+1} = \theta_t + \eta q_t$, where the update direction is masked such that $q_{t,k} = 0$ for all $k \notin S$.

By definition, the selection condition $s_k \geq \tau_{\mathrm{sel}}$ algebraically implies that for any selected parameter, the robustness importance is upper-bounded by the forgetting importance: $I_k^{(\mathrm{adv})} \leq \frac{1}{\tau_{\mathrm{sel}}} I_k^{(\mathrm{f})}$. Since I_k represents the *average* gradient magnitude, we assume this dominance relationship also holds for the *instantaneous* gradients at the current optimization step.

Assumption C.2 (Gradient Alignment in Selected Subspace)**.** We assume that the importance scores derived from dataset-level statistics serve as a valid proxy for the instantaneous gradient magnitudes at step t. Specifically, the condition $I_k^{(\mathrm{adv})} \leq \frac{1}{\tau_{\mathrm{sel}}} I_k^{(\mathrm{f})}$ translates to the following bound on the objective gradients for all active parameters $k \in S$:

$$\left|\nabla_{\theta_k} \mathcal{R}(\theta)\right| \leq \frac{1}{\tau_{\mathrm{sel}}} \left|\nabla_{\theta_k} \mathcal{L}_{\mathrm{unl}}(\theta)\right|. \tag{60}$$

Recall the Assumption 6.1. Mathematically, this smoothness assumption guarantees that the objectives are bounded by a quadratic approximation. For any parameter update $v = \eta q$, we have:

$$\mathcal{L}_{\mathrm{unl}}(\theta + v) \leq \mathcal{L}_{\mathrm{unl}}(\theta) + \nabla \mathcal{L}_{\mathrm{unl}}(\theta)^\top v + \frac{L_F}{2}\|v\|^2, \tag{61}$$

$$\mathcal{R}(\theta + v) \leq \mathcal{R}(\theta) + \nabla \mathcal{R}(\theta)^\top v + \frac{L_R}{2}\|v\|^2. \tag{62}$$

C.3.2 Proof of Theorem 6.5

Proof. We begin by analyzing the change in objectives for a single step t:$v = \eta q_t$. The update direction q_t is constructed to descend $\mathcal{L}_{\mathrm{unl}}$. Thus, for active parameters $k \in S$, $q_{t,k}$ has the opposite sign of $\nabla_{\theta_k} \mathcal{L}_{\mathrm{unl}}(\theta_t)$. Consider the inner product for the robustness objective restricted to S, scaled by the step size η:

$$\begin{aligned} \nabla \mathcal{R}(\theta_t)^\top (\eta q_t) &= \eta \sum_{k\in S} \nabla_{\theta_k} \mathcal{R}(\theta_t) \cdot q_{t,k} \\ &\leq \eta \sum_{k\in S} \left|\nabla_{\theta_k} \mathcal{R}(\theta_t)\right| \cdot \left|q_{t,k}\right|. \end{aligned} \tag{63}$$

Applying Assumption C.2 ($|\nabla_{\theta_k}\mathcal{R}| \leq \frac{1}{\tau_{\text{sel}}}|\nabla_{\theta_k}\mathcal{L}_{\text{unl}}|$):

$$\begin{aligned}\nabla\mathcal{R}(\theta_t)^\top(\eta q_t) &\leq \frac{\eta}{\tau_{\text{sel}}}\sum_{k\in S}|\nabla_{\theta_k}\mathcal{L}_{\text{unl}}(\theta_t)|\cdot|q_{t,k}| \\ &= \frac{\eta}{\tau_{\text{sel}}}\sum_{k\in S}(-\nabla_{\theta_k}\mathcal{L}_{\text{unl}}(\theta_t)\cdot q_{t,k}) \\ &= \frac{1}{\tau_{\text{sel}}}\left(-\nabla\mathcal{L}_{\text{unl}}(\theta_t)^\top(\eta q_t)\right). \end{aligned} \tag{64}$$

Substituting the update vector $v = \eta q_t$ to Eq. (61) and rearranging to bound the negative gradient term:

$$-\nabla\mathcal{L}_{\text{unl}}(\theta_t)^\top(\eta q_t) \leq \mathcal{L}_{\text{unl}}(\theta_t) - \mathcal{L}_{\text{unl}}(\theta_{t+1}) + \frac{L_F\eta^2}{2}\|q_t\|^2. \tag{65}$$

Similarly, according to Eq. (62), for the robustness risk, we have:

$$\mathcal{R}(\theta_{t+1}) - \mathcal{R}(\theta_t) \leq \nabla\mathcal{R}(\theta_t)^\top(\eta q_t) + \frac{L_R\eta^2}{2}\|q_t\|^2. \tag{66}$$

Substituting Eq. (64) and then Eq. (65) into the robustness inequality:

$$\begin{aligned}\mathcal{R}(\theta_{t+1}) - \mathcal{R}(\theta_t) &\leq \frac{1}{\tau_{\text{sel}}}\left(-\nabla\mathcal{L}_{\text{unl}}(\theta_t)^\top(\eta q_t)\right) + \frac{L_R\eta^2}{2}\|q_t\|^2 \\ &\leq \frac{1}{\tau_{\text{sel}}}\left(\mathcal{L}_{\text{unl}}(\theta_t) - \mathcal{L}_{\text{unl}}(\theta_{t+1}) + \frac{L_F\eta^2}{2}\|q_t\|^2\right) + \frac{L_R\eta^2}{2}\|q_t\|^2 \\ &= \frac{1}{\tau_{\text{sel}}}\left(\mathcal{L}_{\text{unl}}(\theta_t) - \mathcal{L}_{\text{unl}}(\theta_{t+1})\right) + \underbrace{\left(\frac{L_R}{2} + \frac{L_F}{2\tau_{\text{sel}}}\right)}_{C_{\text{sel}}}\eta^2\|q_t\|^2. \end{aligned} \tag{67}$$

Summing the inequality over $t = 0, \ldots, T-1$:

$$\sum_{t=0}^{T-1}(\mathcal{R}(\theta_{t+1}) - \mathcal{R}(\theta_t)) \leq \frac{1}{\tau_{\text{sel}}}\sum_{t=0}^{T-1}(\mathcal{L}_{\text{unl}}(\theta_t) - \mathcal{L}_{\text{unl}}(\theta_{t+1})) + C_{\text{sel}}\sum_{t=0}^{T-1}\eta^2\|q_t\|^2. \tag{68}$$

Equivalently, we have:

$$\mathcal{R}(\theta_{\text{unl}}^{\text{sel}}) - \mathcal{R}(\theta_{\text{ori}}) \leq \frac{1}{\tau_{\text{sel}}}\left(\mathcal{L}_{\text{unl}}(\theta_{\text{ori}}) - \mathcal{L}_{\text{unl}}(\theta_{\text{unl}}^{\text{sel}})\right) + C_{\text{sel}}\eta^2\sum_{t=0}^{T-1}\|q_t\|^2. \tag{69}$$

Finally, we bound the accumulation of the quadratic error terms. Recall from Lemma C.1 that the dense update direction $\hat{q}_t$ is norm-bounded, satisfying $\|\hat{q}_t\|^2 \leq B^2$. As our method applies a binary mask M , the squared norm of the masked update direction $q_t = M \odot \hat{q}_t$ is bounded by scaling the full norm limit by the density ratio ρ:

$$\|q_t\|^2 = \|M \odot \hat{q}_t\|^2 \approx \rho\|\hat{q}\|^2 \leq \rho B^2. \tag{70}$$

Substituting $\Delta\mathcal{R}_{\text{sel}}$, $\Delta\mathcal{L}_{\text{unl}}$, and this norm bound into the derived inequality yields the final result:

$$\Delta\mathcal{R}_{\text{sel}} \leq \frac{1}{\tau_{\text{sel}}}\Delta\mathcal{L}_{\text{unl}} + T\cdot C_{\text{sel}}\cdot(\rho\eta^2B^2). \tag{71}$$

This completes the proof. □

C.3.3 Proof of Proposition 6.6

Proof. We compare the robustness degradation bound of our method (Theorem 6.2) with that of the baseline standard unlearning.

Baseline Analysis: For the standard GU, all parameters are updated without selection. This corresponds to a selection ratio of $\rho = 1$ and an importance threshold determined by the worst-case

parameter: $\tau_{\min} = \min_k s_k$. Consequently, the baseline update rule is $\theta_{t+1} = \theta_t + \eta q_t^{\text{base}}$. Similarly, we have:

$$\mathcal{R}(\theta_{t+1}) - \mathcal{R}(\theta_t) \leq \frac{1}{\tau_{\min}}(\mathcal{L}_{\text{unl}}(\theta_t) - \mathcal{L}_{\text{unl}}(\theta_{t+1})) + C_{\text{base}}\eta^2\|q_t^{\text{base}}\|^2, \tag{72}$$

where $C_{\text{base}} \coloneqq \frac{L_R}{2} + \frac{L_F}{2\tau_{\min}}$. As the direction norm is bounded by $\|q_t^{\text{base}}\|^2 \leq B^2$, following a similar derivation, we define the upper bound for the baseline as:

$$\text{Bound}(\Delta\mathcal{R}_{\text{base}}) \coloneqq \frac{1}{\tau_{\min}}\Delta\mathcal{L}_{\text{unl}} + T \cdot C_{\text{base}} \cdot \eta^2 B^2. \tag{73}$$

Comparison: Recall the bound for our Importance-Guided Parameter Selection from Theorem 6.2:

$$\text{Bound}(\Delta\mathcal{R}_{\text{sel}}) = \frac{1}{\tau_{\text{sel}}}\Delta\mathcal{L}_{\text{unl}} + T \cdot C_{\text{sel}} \cdot (\rho\eta^2 B^2). \tag{74}$$

Comparing the bounds, the selection properties ($\tau_{\text{sel}} > \tau_{\min}$ and $\rho < 1$) guarantee:

$$\text{Linear Term:} \quad \frac{1}{\tau_{\text{sel}}} < \frac{1}{\tau_{\min}}, \tag{75}$$

$$\text{Quadratic Term:} \quad TC_{\text{sel}}(\rho\eta^2 B^2) < TC_{\text{base}}(\eta^2 B^2). \tag{76}$$

The second inequality holds because both the smoothness constant is reduced ($C_{\text{sel}} < C_{\text{base}}$) and the sparsity factor scales the norm ($\rho < 1$). Summing these inequalities yields a strictly tighter total bound:

$$\text{Bound}(\Delta\mathcal{R}_{\text{sel}}) < \text{Bound}(\Delta\mathcal{R}_{\text{base}}). \tag{77}$$

This completes the proof. □

C.4 Detailed Theoretical Complexity Analysis

In this section, we analyze the time and space complexity of the proposed GUARD framework. Let d denote the number of model parameters, m the number of anchors in the adversarially vulnerable set $\mathcal{D}_v$, and B the batch size used for unlearning updates. We assume the cost of a standard backpropagation step is $\mathcal{C}_{\text{bp}} \approx \mathcal{O}(d)$.

C.4.1 Time Complexity

The initialization phase involves constructing the vulnerable set $\mathcal{D}_v$ and computing the importance mask. Both operations incur negligible overhead compared to the full unlearning process. Constructing $\mathcal{D}_v$ requires performing a constant number of PGD steps T on a small subset of samples ($m \ll |\mathcal{D}_r|$), resulting in a cost of $\mathcal{O}(m \cdot T \cdot d)$. Similarly, the Importance-Guided Parameter Selection requires a single pass over the forget and vulnerable sets to accumulate gradients, followed by a sorting operation. The complexity is dominated by $\mathcal{O}(|\mathcal{D}_f| \cdot d)$, which is equivalent to the computational cost of a single training epoch on the forget set.

During the unlearning phase, the primary computational cost stems from the Robustness-Aware Gradient Projection. A naive implementation requires computing the Gram matrix $S = JPJ^\top$ directly from the high-dimensional Jacobian $J \in \mathbb{R}^{m \times d}$. This operation involves pairwise dot products of gradients, leading to a complexity of $\mathcal{O}(m^2 d)$. As the model dimension d increases, this quadratic dependence on m scaled by d can become computationally prohibitive.

By employing the Johnson-Lindenstrauss (JL) approximation as described in Section 5.2 to project the anchor gradients into a lower-dimensional subspace $\mathbb{R}^k$ (where $k \ll d$), we effectively decouple the quadratic interaction terms from the high-dimensional parameter space. Consequently, the dominant per-step computational complexity is reduced from $\mathcal{O}(m^2 d)$ to $\mathcal{O}(\mathbf{mkd})$. Since k is independent of d (typically $k \propto \log m$) and m is a small constant, this projection introduces only a marginal linear overhead over standard gradient ascent.

C.4.2 Space Complexity

Regarding memory consumption, GUARD incurs minimal additional storage. The importance mask is a binary vector requiring only $\mathcal{O}(d)$ bits. For the gradient projection, while a naive approach would necessitate storing the full Jacobian matrix ($\mathcal{O}(md)$), our JL-based implementation only requires storing the compressed gradient vectors in $\mathbb{R}^{m \times k}$ and the random seed for the projection matrix. This reduces the space complexity to $\mathcal{O}(\mathbf{mk})$. Given that $mk \ll d$, the memory overhead is effectively zero relative to the model size, facilitating deployment on memory-constrained devices.

C.4.3 Summary and Comparison

We provide a comprehensive comparison of computational costs. To contextualize the efficiency of GUARD, we also analyze the complexity of standard baselines: GA, BS, SalUn, FT and UNNSIR. The methods GA and FT primarily rely on standard first-order gradient updates. Consequently, their per-step time complexity scales linearly with the model dimension as $\mathcal{O}(d)$. Each step of BS consists of a standard forward/backward update, and the extra operations for generating an perturbed example and predicting its label are also $\mathcal{O}(d)$, so the order remains $\mathcal{O}(d)$ (only the constant factor increases). These baselines require no significant initialization overhead or additional memory requirements beyond standard backpropagation buffers ($\mathcal{O}(1)$)

UNSIR requires an initialization phase that generates N_{noise} noise vectors via T_{noise} steps of gradient-based optimization to simulate the forget data, resulting in a computational complexity of $\mathcal{O}(T_{\text{noise}}N_{\text{noise}}d)$. Each impair and repair step is a standard gradient descent update, costing $\mathcal{O}(d)$ per step. Storing all noise matrices requires $\mathcal{O}(N_{\text{noise}} \cdot P_{\text{noise}})$ memory, where P_{noise} denotes the dimensionality of a noise matrix.

SalUn incorporates a weight saliency selection mechanism. While its iterative unlearning phase remains efficient at $\mathcal{O}(d)$, it typically necessitates an initialization pass over the forget set to compute gradient-based sensitivity metrics. This results in a one-time initialization cost of $\mathcal{O}(|\mathcal{D}_f|d)$ and a linear space overhead of $\mathcal{O}(d)$ to store the saliency mask.

MUter comprises three stages: **(Stage I)** performs a one-time preprocessing pass over the training set to construct cached second-order Hessian statistics; **(Stage II)** performs unlearning using the forget samples and the cached Hessian by solving a block linear system; and **(Stage III)** updates the cached statistics to support subsequent unlearning requests. Accordingly, the total cost consists of a dataset-dependent one-time preprocessing cost and a per-request cost, governed by the parameter dimension d, perturbation dimension P_{muter}, the number of linear-solver iterations T_{muter}, and the Neumann-series truncation order K_{muter}. As shown in Table A1, Stage I requires $\mathcal{O}\left(n(d^2 + K_{\text{muter}}P_{\text{muter}}^2 d + P_{\text{muter}}d^2)\right)$ time to construct the cached total Hessian, assuming dense matrix operations and recursive Neumann-series evaluation. The persistent cache requires $\mathcal{O}(d^2 + d)$ space. For each single-sample unlearning request, Stage II computes the necessary derivatives and iteratively solves a $(d + P_{\text{muter}})$-dimensional block linear system (e.g., via conjugate gradient or fixed-point iterations), requiring $\mathcal{O}\left(T_{\text{muter}}(d + P_{\text{muter}})^2\right)$ time for the linear solve. Stage III computes the sample-specific total-Hessian contribution using the Neumann-series approximation and updates the cached statistics, requiring $\mathcal{O}\left(d^2 + K_{\text{muter}}P_{\text{muter}}^2 d + P_{\text{muter}}d^2\right)$ time. These bounds characterize dense matrix operations and exclude model-dependent derivative computation and adversarial perturbation generation costs.

The total complexity of GUARD consists of a one-time initialization phase and the iterative unlearning phase. The **initialization overhead** includes: (1) **Adversarially Vulnerable Set Construction**, which scales as $\mathcal{O}(mTd)$ (where T is the number of PGD steps); and (2) **Importance-Guided Parameter Selection**, which requires a single pass over the forget set, scaling as $\mathcal{O}(|\mathcal{D}_f|d)$. Notably, these initialization costs are equivalent to approximately one standard training epoch and are amortized over the entire unlearning process. As shown in the Table A2, GUARD shares the initialization characteristics of saliency-based methods like SalUN but maintains highly efficient per-step updates via JL projection, demonstrating scalability and efficiency.

Table A1: **Theoretical complexity of MUter across three stages.** Here, $|\mathcal{D}|$ is the number of training samples; d is the parameter dimension; P_{muter} is the perturbation dimension; K_{muter} is the Neumann-series truncation order; T_{muter} is the number of linear-solver iterations. The analysis considers dense matrix operations, excluding derivative computation and adversarial example generation. Space complexity refers to persistent cache storage.

Stage	**Time Complexity**	**Space Complexity**
Pre-Unlearning	$\mathcal{O}\left(\|\mathcal{D}\|(d^2 + K_{\text{muter}}P_{\text{muter}}^2 d + P_{\text{muter}}d^2)\right)$	$\mathcal{O}(d^2 + d)$
Unlearning	$\mathcal{O}\left(T_{\text{muter}}(d + P_{\text{muter}})^2\right)$	$\mathcal{O}(d^2 + d)$
Post-Unlearning	$\mathcal{O}\left(d^2 + K_{\text{muter}}P_{\text{muter}}^2 d + P_{\text{muter}}d^2\right)$	$\mathcal{O}(d^2 + d)$

Table A2: **Theoretical complexity analysis comparing GUARD with baselines.** Here, d is the model dimension, $|\mathcal{D}_f|$ is the forget set size, m is the number of anchors ($m \ll |\mathcal{D}_f|$), T is the PGD steps, k is the JL projection dimension ($k \ll d$), N_{noise} is the number of noise vectors, P_{noise} is the dimensionality of a noise matrix , and T_{noise} is the iteration numbers for noise generation.

Method	**Initialization Cost** (Time)	**Per-Step Cost** (Time)	**Space Overhead**
GA / FT / BS	–	$\mathcal{O}(d)$	$\mathcal{O}(1)$
SalUn	$\mathcal{O}(\lvert\mathcal{D}_f\rvert d)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$
UNSIR	$\mathcal{O}(T_{\text{noise}} N_{\text{noise}} d)$	$\mathcal{O}(d)$	$\mathcal{O}(N_{\text{noise}} \cdot P_{\text{noise}})$
GUARD (Naive)	$\mathcal{O}(\lvert\mathcal{D}_f\rvert d + mTd)$	$\mathcal{O}(m^2 d)$	$\mathcal{O}(md + d)$
GUARD (w/ JL)	$\mathcal{O}(\lvert\mathcal{D}_\mathbf{f}\rvert \mathbf{d} + \mathbf{mTd})$	$\mathcal{O}(\mathbf{mkd})$	$\mathcal{O}(\mathbf{mk} + \mathbf{d})$

D Extended Experiments

D.1 Experimental Settings

In this section, we provide the detailed experimental settings. Our experiments are all conducted using Python 3.8.

Original Training and Retraining-From-Scratch Both the Original model and the Retrain oracle are obtained via FGSM adversarial training [18]. For the original and retrained models trained on the same dataset and model architecture, we used identical adversarial training hyperparameters to ensure a fair comparison. We set the perturbation radius to $\epsilon = 0.02$ for CIFAR-10 and CIFAR-100, and $\epsilon = 0.01$ for Tiny ImageNet. For optimization, we use SGD with momentum 0.9 for CNNs (AllCNN and ResNet-18) and Adam with weight decay 1×10^{-4} for ViT, with a batch size of 64. Specifically, the ViT is configured with a patch size of 4, depth of 6, and 8 attention heads. The filter width of AllCNN is set to 50% for CIFAR-10 and 100% for CIFAR-100. On CIFAR-10, AllCNN is trained for 30 epochs with an initial learning rate (lr) of 0.1, while ResNet-18 is trained for 100 epochs with lr=0.01. On CIFAR-100, both architectures follow the same schedule of 100 epochs with an initial lr of 0.1. On Tiny ImageNet, to ensure convergence, we extend training to 200 epochs, setting the initial lr to 0.05 for AllCNN and 0.025 for ResNet-18. On Tiny ImageNet, to ensure stable convergence, we adopt an enhanced training recipe for CNNs: we employ SGD with Nesterov momentum, weight decay 5×10^{-4}, Cross-Entropy loss with label smoothing (0.1), and a Cosine Annealing scheduler with a 5-epoch linear warmup. The AllCNN filter width is increased to 150% for this dataset.

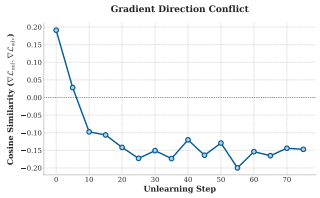
Hyperparameters for GUARD We set the parameter selection ratio to $\rho = 0.0001$ (0.01%) across all baselines, with the exception of FT where $\rho = 0.001$ is adopted. For the anchor mechanism, we consistently use $m = 32$ anchors and refresh their gradients every 5 optimization steps. Regarding the construction of simulated $\mathcal{D}_v$, we employ a relatively large perturbation radius (e.g., $\epsilon = 40/255$). This magnitude is chosen to ensure that samples from the forget set $\mathcal{D}_f$ can be successfully perturbed across the decision boundary to align with the retain class distributions. When generating $\mathcal{D}_v$ from $\mathcal{D}_r$, we employ a single mini-batch of 64 samples as a representative subset, rather than using the entire dataset. For the JL approximation, we set the dimension to $k = 512$.

D.2 Experiments of Gradient Direction Conflict

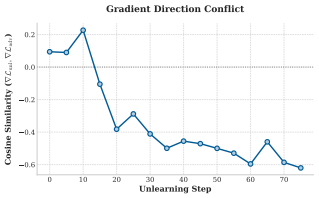
We validate the Gradient Direction Conflict phenomenon described in the main text Section 4 through a broader range of attribution experiments. Specifically, we plot the gradient cosine similarity during the unlearning process for three baselines—Gradient Ascent (GA), Random Label (RL), and Fine-tuning (FT)—across all four settings. As illustrated in Fig. A1, the experimental results show that the cosine similarity between the unlearning gradient $\nabla_\theta \mathcal{L}_{\text{unl}}$ and the robustness gradient $\nabla_\theta \mathcal{L}_{\text{adv}}$ remains significantly negative throughout the majority of unlearning steps. This high consistency across different architectures and datasets demonstrates that the conflict between unlearning and robustness objectives is not an isolated result of specific hyperparameters. Instead, it is intrinsic that unlearning inevitably induces a destructive shift in the established robust boundaries.

Notably, we observe that when fine-tuning the model using only the retain set, the gradientg direction conflict is relatively marginal. In contrast, unlearning methods that directly utilize the forget set exhibit more pronounced gradient conflicts, regardless of the unlearning loss functions employed. This disparity may arises because fine-tuning on the retain set encourages the model to stay within the smooth regions of the loss landscape, thus maintaining a higher degree of alignment with robustness objectives. Conversely, methods targeting the forget set often induce drastic parameter shifts, which inevitably disrupts the delicate robust boundaries.

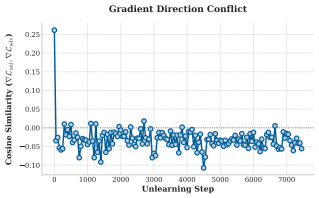
Furthermore, the absolute negative value of the cosine similarity exhibits an increasing trend as unlearning proceeds. This reveals an inherent trade-off between the unlearning task and robustness preservation; notably, this conflict intensifies as the degree of unlearning deepens over time.



(a) GA with AllCNN on CIFAR-10



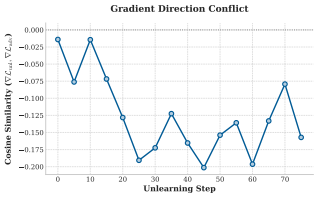
(b) RL with AllCNN on CIFAR-10



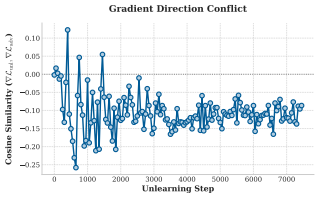
(c) FT with AllCNN on CIFAR-10



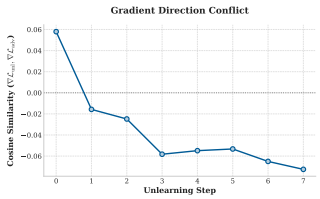
(d) GA with ResNet-18 on CIFAR-10



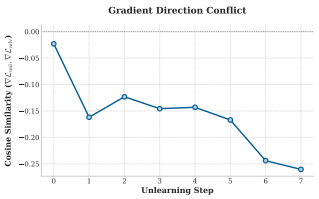
(e) RL with ResNet-18 on CIFAR-10



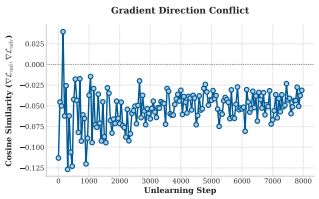
(f) FT with ResNet-18 on CIFAR-10



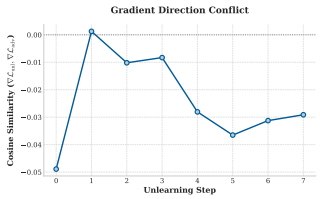
(g) GA with AllCNN on CIFAR-100



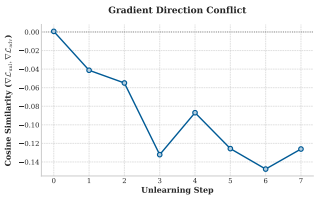
(h) RL with AllCNN on CIFAR-100



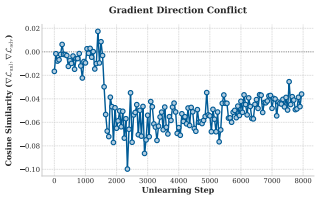
(i) FT with AllCNN on CIFAR-100



(j) GA with ResNet-18 on CIFAR-100



(k) RL with ResNet-18 on CIFAR-100

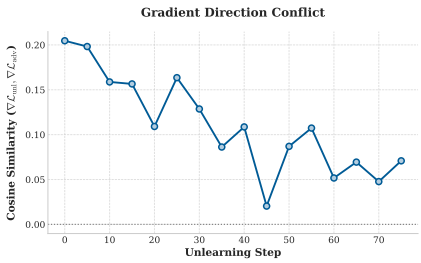


(l) FT with ResNet-18 on CIFAR-100

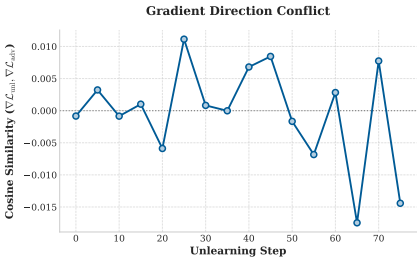
Figure A1: Gradient Direction Conflict experiments for baselines: gradient cosine similarity trend during different GU methods.

We also demonstrate the gradient cosine similarity curves of the models unlearned using GUARD-enhanced baselines (GA, RL and FT). As shown in Fig. A2, the application of our proposed GUARD significantly mitigates, and in some cases entirely eliminates, the Gradient Direction Conflict. Specif-

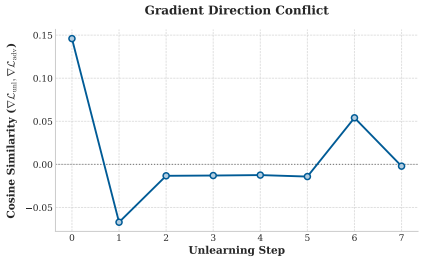
ically, the cosine similarity between the unlearning and robustness gradients largely shifts from the negative region to near-zero or even positive values across different settings. This transition confirms that GUARD effectively decouples the unlearning objective from the adversarial robustness objective,by successfully removing the robustness-harmful components from unlearning updates.



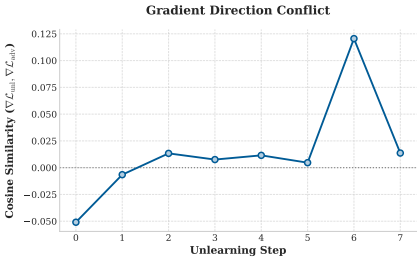
(a) GA+GUARD with AllCNN on CIFAR-10



(b) GA+GUARD with ResNet-18 on CIFAR-10

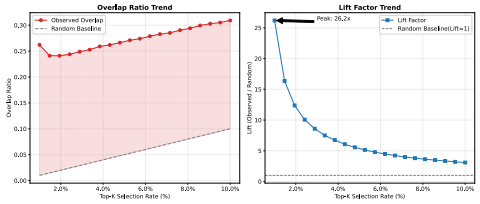


(c) GA+GUARD with AllCNN on CIFAR-100

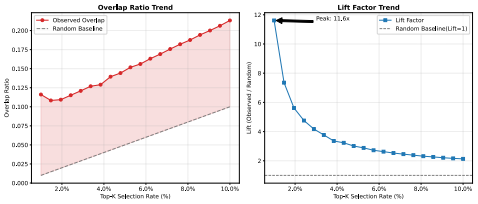


(d) GA+GUARD with ResNet-18 on CIFAR-100

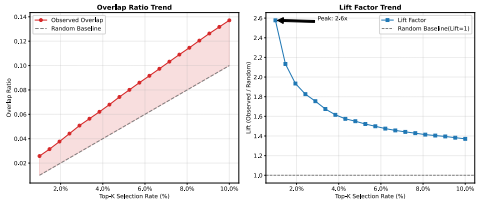
Figure A2: Gradient Direction Conflict Experiments for GUARD-enhanced baselines: gradient cosine similarity trend during different GU methods.



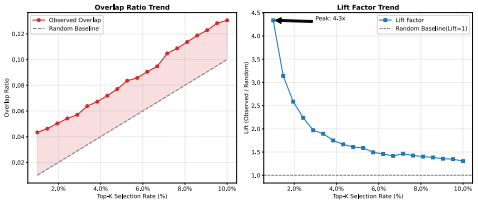
(a) AllCNN on CIFAR-10



(b) AllCNN on CIFAR-100



(c) ResNet-18 on CIFAR-10



(d) ResNet-18 on CIFAR-100

Figure A3: Parameter Entanglement analysis across different architectures and datasets. For each subfigure, left column shows the Overlap Ratio Trend, and right column shows the Lift Factor Trend.

D.3 Experiments of Parameter Entanglement

The comprehensive results across all settings further substantiate the Parameter Entanglement phenomenon stated in the main text Section 4, as shown in Fig. A3. By measuring the lift factor across various thresholds k, we consistently find that the most critical parameters for unlearning and robustness are highly non-independent, with the overlap significantly exceeding the $k\%$ random baseline—reaching a maximum lift factor of $26.2\times$.

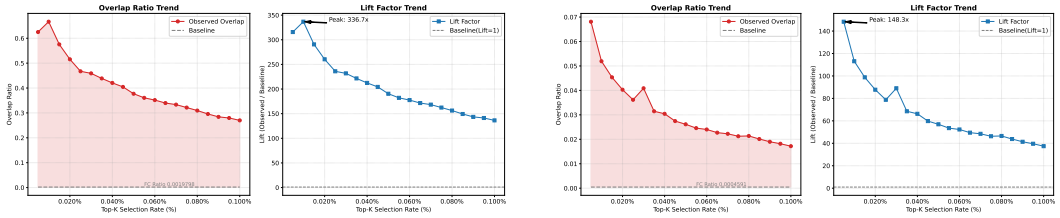
Additionally, we observe that the Parameter Entanglement is more pronounced in lightweight models. Specifically, the lift factor calculated for AllCNN is significantly higher than that of ResNet-18 across all experimental settings. This suggests that limited model capacity forces a higher degree of parameter sharing between different tasks: as the total number of parameters decreases, the model is compelled to utilize the same high-importance weights for both unlearning and robustness preservation. This observation makes models more susceptible to robustness degradation during the unlearning process.

D.4 Unlearning with Fully Connected (FC) Layer Only

Our further analysis show that important parameters with high ratio score s exhibit significant enrichment within the Fully Connected (FC) layers. As shown in Table A3, despite constituting only $\approx 0.2\%$ of the total network parameters (denoted as the *Baseline*), FC parameters comprise the majority (often $> 50\%$) of the top-ranked important subset. Following a similar experimtental design in Section 4, we plot the overlap ratio as a function of the selection ratio ρ. To better quantify this alignment, we define the **Lift Factor** as the ratio of the observed overlap to the natural baseline. As shown in Fig. A4, we observe that both the overlap ratio and the Lift Factor rises significantly as ρ decreases, peaking at $336.7\times$ over the baseline. These results confirm that as the selection ratio narrows, parameter importance effectively collapses toward high-level semantic decision regions.

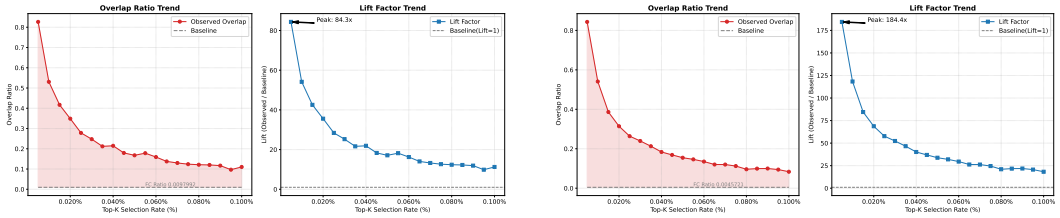
Table A3: Comparison of the proportion of parameters belonging to the Fully Connected (FC) layer between the whole network (Baseline) and the subset selected by our method ($\rho = 1 \times 10^{-4}$).

Model	Dataset	Baseline (FC / Total)	Selected (FC / Subset)
AllCNN	CIFAR-10	0.20%	66.67%
AllCNN	CIFAR-100	0.20%	53.57%
ResNet-18	CIFAR-100	0.05%	51.78%



(a) AllCNN on CIFAR-10

(b) ResNet-18 on CIFAR-10



(c) AllCNN on CIFAR-100

(d) ResNet-18 on CIFAR-100

Figure A4: Fully Connected (FC) layer parameter overlap analysis across settings. For each subfigure, left column shows the Overlap Ratio trend, and right column shows the Lift Factor trend.

Consequently, we propose applying the GUARD framework exclusively to the FC layer (Classifier) to enhance efficiency. In this way, our Robustness-Aware Gradient Projection avoids the calculation of full-parameter gradients for both robustness and unlearning loss. As shown in Table A4, this *Head-Only* strategy achieves unlearning effectiveness and robustness preservation close to full-parameter update version.

Table A4: Class unlearning results using only FC layer on CIFAR-10 with AllCNN.

	Methods	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\mathrm{RA}_r \uparrow$	Avg. Gap
	Original	$68.80_{\pm 0.52}$ (68.80)	$91.62_{\pm 0.14}$ (3.52)	$81.00_{\pm 0.33}$ (1.41)	$71.45_{\pm 0.29}$ (2.37)	19.03
	Retrain	$0.00_{\pm 0.00}$ (0.00)	$95.14_{\pm 0.12}$ (0.00)	$82.41_{\pm 0.21}$ (0.00)	$73.82_{\pm 0.45}$ (0.00)	0.00
	Muter	$9.80_{\pm 0.64}$ (9.80)	$53.89_{\pm 1.23}$ (41.25)	$46.36_{\pm 0.88}$ (36.05)	$30.30_{\pm 1.05}$ (43.52)	32.66
w/o D_r	GA	$0.39_{\pm 0.12}$ (0.39)	$84.28_{\pm 0.41}$ (10.86)	$75.87_{\pm 0.35}$ (6.54)	$55.86_{\pm 1.02}$ (17.96)	8.94
w/o D_r	**GA+ours**	$0.52_{\pm 0.15}$ (0.52)	$92.44_{\pm 0.28}$ (2.70)	$81.21_{\pm 0.31}$ (1.20)	$74.97_{\pm 0.44}$ (-1.15)	0.82
w/o D_r	BS	$0.42_{\pm 0.11}$ (0.42)	$81.60_{\pm 0.55}$ (13.54)	$72.32_{\pm 0.62}$ (10.09)	$57.37_{\pm 1.15}$ (16.45)	10.13
w/o D_r	**BS+ours**	$0.50_{\pm 0.18}$ (0.50)	$91.80_{\pm 0.24}$ (3.34)	$81.51_{\pm 0.29}$ (0.90)	$74.12_{\pm 0.38}$ (-0.30)	1.11
w/o D_r	SalUn	$0.46_{\pm 0.09}$ (0.46)	$86.38_{\pm 0.47}$ (8.76)	$77.09_{\pm 0.51}$ (5.32)	$59.91_{\pm 0.98}$ (13.91)	7.11
w/o D_r	**SalUn+ours**	$0.48_{\pm 0.14}$ (0.48)	$92.11_{\pm 0.26}$ (3.03)	$81.39_{\pm 0.33}$ (1.02)	$74.59_{\pm 0.41}$ (-0.77)	0.94
w/ D_r	FT	$0.00_{\pm 0.00}$ (0.00)	$98.95_{\pm 0.08}$ (-3.81)	$86.74_{\pm 0.22}$ (-4.33)	$28.17_{\pm 1.45}$ (45.65)	9.38
w/ D_r	**FT+ours**	$0.00_{\pm 0.00}$ (0.00)	$93.39_{\pm 0.25}$ (1.75)	$82.86_{\pm 0.30}$ (-0.45)	$73.05_{\pm 0.52}$ (0.77)	0.52
w/ D_r	UNSIR	$0.00_{\pm 0.00}$ (0.00)	$81.03_{\pm 0.61}$ (14.11)	$77.28_{\pm 0.58}$ (5.13)	$15.06_{\pm 1.34}$ (58.76)	19.50
w/ D_r	**UNSIR+ours**	$0.00_{\pm 0.00}$ (0.00)	$93.27_{\pm 0.21}$ (1.87)	$82.52_{\pm 0.27}$ (-0.11)	$74.50_{\pm 0.46}$ (-0.68)	**0.27**

We attribute this to the fact that class unlearning primarily relies on reshaping decision boundaries, which is achievable via the linear classifier (FC layer) without perturbing the robust feature extractor. In the FC-only setting, the lower-dimensional parameter space ($|W| \ll |\theta|$) allows gradient projection to more effectively resolve conflicts between unlearning and robustness objectives. Additionally, freezing the backbone minimizes gradient noise and stabilizes optimization, offering an optimal trade-off by performing unlearning at the decision level while preserving robust feature representations. Futhermore, this strategy reduces the optimization complexity from $O(|\theta|)$ to $O(|W|)$, significantly reducing the computational and memory overheads for backpropagation and projection operations. Fig. A5 provides a quantitative comparison of the running times.

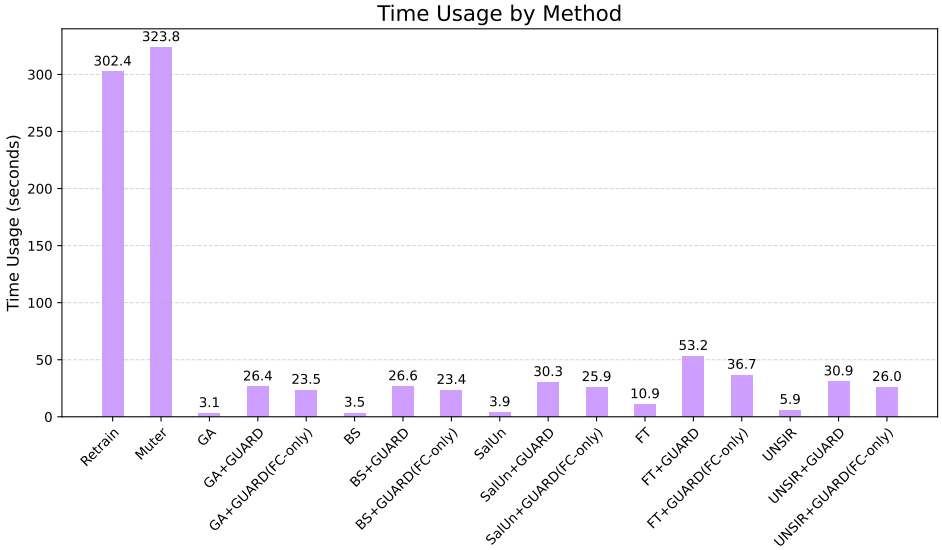


Figure A5: Running time for baselines and GUARD-enhanced version, as well as the FC-only version, with AllCNN on CIFAR-10.

D.5 Extended Class Unlearning Results

D.5.1 Class Unlearning on Varying Architectures and Datasets.

This section presents additional experimental results for class unlearning conducted on other datasets and model architectures. Table A6 demonstrates the results using CIFAR-100 dataset, while Table A7 showing the results in Tiny Imagenet dataset. Note that 'Aquarium Fish' (class with index 1) is randomly choosen as the target unlearning class for CIFAR-100, and 'European Fire Salamander' (class with index 1) is the target class for Tiny ImageNet.

These results consistently demonstrate that our method achieves a state-of-the-art performance in terms of both unlearning efficacy and robustness preservation across diverse settings. This highlights the strong scalability of our proposed framework.

Table A5: Class unlearning (Class-3) results with ViT on CIFAR-10.

	Methods	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\text{RA}_r \uparrow$	Avg. Gap
	Original	$78.14_{\pm 0.02}$ (78.14)	$91.93_{\pm 0.02}$ (0.21)	$78.14_{\pm 0.04}$ (0.24)	$57.46_{\pm 0.04}$ (0.43)	19.76
	Retrain	$0.00_{\pm 0.00}$ (0.00)	$92.14_{\pm 0.02}$ (0.00)	$78.38_{\pm 0.03}$ (0.00)	$57.89_{\pm 0.05}$ (0.00)	0.00
	Muter	$1.06_{\pm 0.09}$ (1.06)	$85.58_{\pm 0.23}$ (6.56)	$74.87_{\pm 0.65}$ (3.51)	$46.33_{\pm 1.04}$ (11.56)	5.67
w/o D_r	GA	$1.08_{\pm 0.14}$ (1.08)	$90.86_{\pm 0.31}$ (1.28)	$78.74_{\pm 0.35}$ (-0.36)	$54.14_{\pm 1.02}$ (3.75)	1.44
w/o D_r	**GA+ours**	$1.94_{\pm 0.45}$ (1.94)	$92.74_{\pm 0.28}$ (-0.60)	$80.02_{\pm 0.31}$ (-1.64)	$58.52_{\pm 0.50}$ (-0.63)	-0.23
w/o D_r	BS	$0.34_{\pm 0.06}$ (0.34)	$87.21_{\pm 0.05}$ (4.93)	$76.81_{\pm 0.11}$ (1.57)	$50.89_{\pm 0.09}$ (7.00)	3.46
w/o D_r	**BS+ours**	$0.36_{\pm 0.08}$ (0.36)	$89.80_{\pm 0.10}$ (2.34)	$78.16_{\pm 0.09}$ (0.22)	$58.23_{\pm 0.38}$ (-0.34)	0.65
w/o D_r	SalUn	$0.56_{\pm 0.05}$ (0.56)	$81.48_{\pm 0.17}$ (10.66)	$71.61_{\pm 0.31}$ (6.77)	$46.92_{\pm 0.12}$ (10.97)	7.24
w/o D_r	**SalUn+ours**	$0.82_{\pm 0.09}$ (0.82)	$87.18_{\pm 0.22}$ (4.96)	$75.37_{\pm 0.13}$ (3.01)	$55.59_{\pm 0.18}$ (2.30)	2.77
w/ D_r	FT	$6.56_{\pm 0.06}$ (6.56)	$97.15_{\pm 0.08}$ (-5.01)	$83.72_{\pm 0.12}$ (-5.34)	$56.58_{\pm 0.06}$ (1.31)	-0.62
w/ D_r	**FT+ours**	$6.46_{\pm 0.06}$ (6.46)	$97.10_{\pm 0.05}$ (-4.96)	$82.86_{\pm 0.10}$ (-4.48)	$58.50_{\pm 0.08}$ (-0.61)	**-0.90**
w/ D_r	UNSIR	$0.50_{\pm 0.00}$ (0.50)	$95.23_{\pm 0.11}$ (-3.09)	$83.19_{\pm 0.28}$ (-4.81)	$43.48_{\pm 0.64}$ (14.41)	1.75
w/ D_r	**UNSIR+ours**	$1.48_{\pm 0.18}$ (1.48)	$87.97_{\pm 0.21}$ (4.17)	$75.20_{\pm 0.80}$ (3.18)	$54.50_{\pm 0.11}$ (3.39)	3.06

Table A6: Class unlearning (class-1) results on CIFAR-100.

	Methods	AllCNN					ResNet-18				
		$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\text{RA}_r \uparrow$	Avg. Gap	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\text{RA}_r \uparrow$	Avg. Gap
	Original	$100.0_{\pm 0.00}$ (100.0)	$99.96_{\pm 0.26}$ (0.00)	$52.32_{\pm 0.31}$ (0.11)	$99.86_{\pm 0.01}$ (0.08)	25.05	$100.0_{\pm 0.00}$ (100.0)	$99.97_{\pm 0.01}$ (0.00)	$52.43_{\pm 0.03}$ (0.35)	$99.94_{\pm 0.03}$ (0.00)	25.09
	Retrain	$0.00_{\pm 0.00}$ (0.00)	$99.96_{\pm 0.01}$ (0.00)	$54.14_{\pm 0.06}$ (0.00)	$98.51_{\pm 0.05}$ (0.00)	0.00	$0.00_{\pm 0.00}$ (0.00)	$99.97_{\pm 0.01}$ (0.00)	$52.78_{\pm 0.04}$ (0.00)	$99.94_{\pm 0.02}$ (0.00)	0.00
	Muter	$0.00_{\pm 0.00}$ (0.00)	$64.86_{\pm 2.42}$ (35.10)	$26.39_{\pm 2.53}$ (27.75)	$21.19_{\pm 1.89}$ (77.32)	35.04	$91.60_{\pm 0.40}$ (91.60)	$97.87_{\pm 0.23}$ (2.10)	$44.93_{\pm 1.22}$ (7.85)	$91.41_{\pm 0.74}$ (8.53)	27.52
w/o D_r	GA	$2.20_{\pm 0.41}$ (2.20)	$99.89_{\pm 0.12}$ (0.07)	$49.86_{\pm 0.53}$ (4.28)	$73.01_{\pm 0.39}$ (25.50)	8.01	$0.00_{\pm 0.00}$ (0.00)	$99.72_{\pm 0.08}$ (0.25)	$45.34_{\pm 0.22}$ (7.44)	$79.71_{\pm 0.34}$ (20.23)	6.98
w/o D_r	**GA+ours**	$0.00_{\pm 0.00}$ (0.00)	$99.96_{\pm 0.02}$ (0.00)	$53.30_{\pm 0.14}$ (0.84)	$98.17_{\pm 0.06}$ (0.34)	**0.30**	$0.00_{\pm 0.00}$ (0.00)	$99.97_{\pm 0.01}$ (0.00)	$52.40_{\pm 0.20}$ (0.38)	$99.94_{\pm 0.02}$ (0.00)	0.10
w/o D_r	BS	$1.60_{\pm 0.09}$ (1.60)	$96.08_{\pm 0.33}$ (3.88)	$45.03_{\pm 0.52}$ (9.11)	$42.65_{\pm 0.43}$ (55.86)	17.61	$1.20_{\pm 0.17}$ (1.20)	$95.55_{\pm 0.08}$ (4.42)	$36.51_{\pm 0.16}$ (16.27)	$53.68_{\pm 0.49}$ (46.26)	17.04
w/o D_r	**BS+ours**	$2.20_{\pm 0.09}$ (2.20)	$99.96_{\pm 0.04}$ (0.00)	$52.27_{\pm 0.49}$ (1.87)	$97.99_{\pm 0.05}$ (0.52)	1.15	$0.00_{\pm 0.00}$ (0.00)	$99.97_{\pm 0.28}$ (0.00)	$52.40_{\pm 0.30}$ (0.38)	$99.94_{\pm 0.26}$ (0.00)	0.10
w/o D_r	SalUn	$1.80_{\pm 0.04}$ (1.80)	$97.62_{\pm 0.34}$ (2.34)	$45.68_{\pm 0.33}$ (8.46)	$46.61_{\pm 0.20}$ (51.90)	16.13	$0.60_{\pm 0.24}$ (0.60)	$99.45_{\pm 0.03}$ (0.52)	$45.00_{\pm 0.09}$ (7.78)	$72.93_{\pm 0.07}$ (27.01)	8.98
w/o D_r	**SalUn+ours**	$0.00_{\pm 0.00}$ (0.00)	$99.96_{\pm 0.00}$ (0.00)	$53.25_{\pm 0.03}$ (0.89)	$97.93_{\pm 0.01}$ (0.58)	0.37	$0.00_{\pm 0.00}$ (0.00)	$99.97_{\pm 0.01}$ (0.00)	$52.45_{\pm 0.05}$ (0.33)	$99.92_{\pm 0.00}$ (0.02)	**0.09**
w/ D_r	FT	$0.20_{\pm 0.12}$ (0.20)	$93.42_{\pm 0.97}$ (6.54)	$53.95_{\pm 1.62}$ (0.19)	$10.67_{\pm 1.48}$ (87.84)	23.69	$0.00_{\pm 0.00}$ (0.00)	$98.71_{\pm 0.15}$ (1.26)	$52.73_{\pm 0.21}$ (0.05)	$18.58_{\pm 0.52}$ (81.36)	20.67
w/ D_r	**FT+ours**	$0.20_{\pm 0.06}$ (0.20)	$97.82_{\pm 0.82}$ (2.14)	$47.45_{\pm 1.22}$ (6.69)	$35.77_{\pm 1.35}$ (62.74)	17.94	$0.00_{\pm 0.00}$ (0.00)	$95.32_{\pm 0.37}$ (4.65)	$48.87_{\pm 0.54}$ (3.91)	$39.79_{\pm 1.01}$ (60.15)	17.18
w/ D_r	UNSIR	$1.00_{\pm 0.02}$ (1.00)	$90.56_{\pm 0.53}$ (9.40)	$47.25_{\pm 0.03}$ (6.89)	$24.12_{\pm 0.19}$ (74.39)	22.92	$0.30_{\pm 0.17}$ (0.30)	$87.15_{\pm 1.38}$ (12.82)	$45.49_{\pm 1.06}$ (7.29)	$21.98_{\pm 2.43}$ (77.96)	24.59
w/ D_r	**UNSIR+ours**	$0.00_{\pm 0.00}$ (0.00)	$99.95_{\pm 0.02}$ (0.01)	$53.45_{\pm 0.19}$ (0.69)	$95.41_{\pm 0.04}$ (3.10)	0.95	$0.00_{\pm 0.00}$ (0.00)	$99.97_{\pm 0.01}$ (0.00)	$49.28_{\pm 0.12}$ (3.50)	$99.01_{\pm 0.05}$ (0.93)	1.11

Table A7: Class unlearning (class-1) results on Tiny Imagenet.

	Methods	AllCNN					ResNet-18				
		$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\text{RA}_r \uparrow$	Avg. Gap	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\text{RA}_r \uparrow$	Avg. Gap
	Original	$83.40_{\pm 0.06}$ (83.40)	$89.56_{\pm 0.02}$ (0.14)	$59.42_{\pm 0.08}$ (0.63)	$65.36_{\pm 0.05}$ (0.72)	21.22	$100.0_{\pm 0.00}$ (100.0)	$99.98_{\pm 0.00}$ (0.00)	$57.50_{\pm 0.02}$ (0.41)	$99.96_{\pm 0.01}$ (0.00)	25.10
	Retrain	$0.00_{\pm 0.00}$ (0.00)	$89.70_{\pm 0.05}$ (0.00)	$60.05_{\pm 0.08}$ (0.00)	$66.08_{\pm 0.09}$ (0.00)	0.00	$0.00_{\pm 0.00}$ (0.00)	$99.98_{\pm 0.00}$ (0.00)	$57.91_{\pm 0.02}$ (0.00)	$99.96_{\pm 0.01}$ (0.00)	0.00
	Muter	$5.40_{\pm 0.82}$ (5.40)	$78.32_{\pm 1.24}$ (11.38)	$49.21_{\pm 0.95}$ (10.84)	$43.38_{\pm 1.15}$ (22.70)	12.58	$12.20_{\pm 1.45}$ (12.20)	$41.58_{\pm 2.10}$ (58.40)	$15.06_{\pm 0.88}$ (42.85)	$23.76_{\pm 1.62}$ (76.20)	47.41
w/o D_r	GA	$1.40_{\pm 0.25}$ (1.40)	$70.80_{\pm 1.13}$ (18.90)	$44.82_{\pm 0.76}$ (15.23)	$31.69_{\pm 0.89}$ (34.39)	17.48	$0.00_{\pm 0.00}$ (0.00)	$96.62_{\pm 0.42}$ (3.36)	$46.73_{\pm 0.51}$ (11.18)	$69.75_{\pm 0.68}$ (30.21)	11.19
w/o D_r	**GA+ours**	$6.40_{\pm 0.55}$ (6.40)	$89.29_{\pm 0.28}$ (0.41)	$59.08_{\pm 0.31}$ (0.97)	$64.31_{\pm 0.45}$ (1.77)	2.39	$0.00_{\pm 0.00}$ (0.00)	$99.98_{\pm 0.02}$ (0.00)	$57.51_{\pm 0.26}$ (0.40)	$99.95_{\pm 0.04}$ (0.01)	**0.10**
w/o D_r	BS	$4.00_{\pm 0.61}$ (4.00)	$72.36_{\pm 0.93}$ (17.34)	$45.82_{\pm 0.82}$ (14.23)	$31.83_{\pm 1.04}$ (34.25)	17.46	$0.00_{\pm 0.00}$ (0.00)	$94.54_{\pm 0.58}$ (5.44)	$40.56_{\pm 0.77}$ (17.35)	$59.51_{\pm 1.23}$ (40.45)	15.81
w/o D_r	**BS+ours**	$0.00_{\pm 0.00}$ (0.00)	$89.21_{\pm 0.34}$ (0.49)	$58.81_{\pm 0.29}$ (1.24)	$65.05_{\pm 0.41}$ (1.03)	0.69	$0.80_{\pm 0.22}$ (0.80)	$99.98_{\pm 0.03}$ (0.00)	$57.42_{\pm 0.35}$ (0.49)	$99.96_{\pm 0.02}$ (0.00)	0.32
w/o D_r	SalUn	$1.20_{\pm 0.28}$ (1.20)	$68.34_{\pm 0.87}$ (21.36)	$43.09_{\pm 0.65}$ (16.96)	$28.57_{\pm 0.92}$ (37.51)	19.26	$2.20_{\pm 0.35}$ (2.20)	$96.59_{\pm 0.44}$ (3.39)	$45.60_{\pm 0.69}$ (12.31)	$72.81_{\pm 0.81}$ (27.15)	11.26
w/o D_r	**SalUn+ours**	$0.00_{\pm 0.00}$ (0.00)	$89.48_{\pm 0.21}$ (0.22)	$59.22_{\pm 0.24}$ (0.83)	$65.98_{\pm 0.37}$ (0.10)	**0.29**	$0.00_{\pm 0.00}$ (0.00)	$99.98_{\pm 0.01}$ (0.00)	$57.45_{\pm 0.28}$ (0.46)	$99.95_{\pm 0.03}$ (0.01)	0.12
w/ D_r	FT	$1.60_{\pm 0.38}$ (1.60)	$82.01_{\pm 0.75}$ (7.69)	$57.27_{\pm 0.54}$ (2.78)	$11.98_{\pm 1.12}$ (54.10)	16.54	$0.00_{\pm 0.00}$ (0.00)	$98.13_{\pm 0.32}$ (1.85)	$56.13_{\pm 0.49}$ (1.78)	$35.13_{\pm 1.34}$ (64.83)	17.12
w/ D_r	**FT+ours**	$3.20_{\pm 0.45}$ (3.20)	$87.86_{\pm 0.41}$ (1.84)	$59.73_{\pm 0.33}$ (0.32)	$44.43_{\pm 0.76}$ (21.65)	6.75	$4.40_{\pm 0.56}$ (4.40)	$99.98_{\pm 0.02}$ (0.00)	$58.59_{\pm 0.37}$ (-0.68)	$98.89_{\pm 0.18}$ (1.07)	1.20
w/ D_r	UNSIR	$0.40_{\pm 0.11}$ (0.40)	$81.74_{\pm 0.68}$ (7.96)	$55.57_{\pm 0.59}$ (4.48)	$34.95_{\pm 1.05}$ (31.13)	11.00	$3.00_{\pm 0.49}$ (3.00)	$94.19_{\pm 0.62}$ (5.79)	$54.97_{\pm 0.55}$ (2.94)	$59.68_{\pm 1.10}$ (40.28)	13.00
w/ D_r	**UNSIR+ours**	$0.00_{\pm 0.00}$ (0.00)	$89.40_{\pm 0.26}$ (0.30)	$59.14_{\pm 0.22}$ (0.91)	$65.59_{\pm 0.34}$ (0.49)	0.43	$0.00_{\pm 0.00}$ (0.00)	$99.93_{\pm 0.06}$ (0.05)	$55.13_{\pm 0.31}$ (2.78)	$98.98_{\pm 0.14}$ (0.98)	0.95

D.5.2 Class Unlearning with Different Target Classes

In this section, we present the class unlearning results across different target classes using the AllCNN model on the CIFAR-10 dataset. Specifically, classes with index 1, 5, 7, 8, 9 were randomly selected as the target classes for forgetting. We use GA and SalUN baseline as representative examples.

As illustrated in Table A8, our method demonstrates stability across different target classes, achieving narrow performance gaps relative to the Retrain oracle. This uniformity indicates that our approach ensures reliable unlearning performance and robustness preservation in diverse scenarios.

Table A8: Class unlearning results with different target classes on CIFAR-10 with AllCNN.

	Methods	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\mathrm{RA}_r \uparrow$	Avg. Gap ↓
Class 1	Retrain	$0.00_{\pm 0.00}$ (0.00)	$87.97_{\pm 0.00}$ (0.00)	$75.58_{\pm 0.01}$ (0.00)	$65.96_{\pm 0.02}$ (0.00)	0.00
	GA	$6.26_{\pm 0.17}$ (6.26)	$70.08_{\pm 0.56}$ (17.89)	$62.23_{\pm 0.35}$ (13.35)	$41.52_{\pm 0.48}$ (24.44)	15.48
	GA+ours	$0.00_{\pm 0.00}$ (0.00)	$85.27_{\pm 0.22}$ (2.70)	$74.28_{\pm 0.31}$ (1.30)	$62.51_{\pm 0.48}$ (3.45)	**1.86**
	SalUn	$1.26_{\pm 0.41}$ (1.26)	$74.20_{\pm 0.27}$ (13.77)	$65.88_{\pm 0.46}$ (9.70)	$44.97_{\pm 0.35}$ (20.99)	11.43
	SalUn+ours	$1.54_{\pm 1.13}$ (1.54)	$85.54_{\pm 1.00}$ (2.43)	$74.34_{\pm 0.82}$ (1.24)	$61.78_{\pm 1.07}$ (4.18)	2.35
Class 5	Retrain	$0.00_{\pm 0.00}$ (0.00)	$92.62_{\pm 0.02}$ (0.00)	$80.40_{\pm 0.08}$ (0.00)	$73.00_{\pm 0.05}$ (0.00)	0.00
	GA	$3.86_{\pm 0.67}$ (3.86)	$77.74_{\pm 0.73}$ (14.88)	$70.58_{\pm 0.36}$ (9.82)	$50.11_{\pm 0.22}$ (22.89)	12.86
	GA+ours	$0.10_{\pm 0.01}$ (0.10)	$92.37_{\pm 0.31}$ (0.25)	$80.11_{\pm 0.51}$ (0.29)	$71.68_{\pm 0.53}$ (1.32)	**0.49**
	SalUn	$0.50_{\pm 0.11}$ (0.50)	$80.26_{\pm 0.66}$ (12.36)	$71.22_{\pm 0.58}$ (9.18)	$53.37_{\pm 0.40}$ (19.63)	10.42
	SalUn+ours	$0.46_{\pm 0.18}$ (0.46)	$86.90_{\pm 0.50}$ (5.72)	$77.11_{\pm 1.14}$ (3.29)	$68.61_{\pm 0.49}$ (4.39)	3.46
Class 7	Retrain	$0.00_{\pm 0.00}$ (0.00)	$88.89_{\pm 0.07}$ (0.00)	$77.48_{\pm 0.04}$ (0.00)	$69.18_{\pm 0.09}$ (0.00)	0.00
	GA	$3.71_{\pm 0.38}$ (3.71)	$70.35_{\pm 0.31}$ (18.54)	$65.27_{\pm 0.47}$ (12.21)	$45.06_{\pm 0.35}$ (24.12)	14.64
	GA+ours	$3.76_{\pm 0.24}$ (3.76)	$88.86_{\pm 0.12}$ (0.03)	$77.47_{\pm 0.40}$ (0.01)	$67.26_{\pm 0.35}$ (1.92)	**1.43**
	SalUn	$0.92_{\pm 0.25}$ (0.92)	$80.05_{\pm 0.45}$ (8.84)	$70.17_{\pm 1.02}$ (7.31)	$55.15_{\pm 0.42}$ (14.03)	7.78
	SalUn+ours	$1.92_{\pm 0.30}$ (1.92)	$86.33_{\pm 0.66}$ (2.56)	$75.08_{\pm 1.18}$ (2.40)	$65.56_{\pm 0.40}$ (3.62)	2.63
Class 8	Retrain	$0.00_{\pm 0.00}$ (0.00)	$89.06_{\pm 0.07}$ (0.00)	$76.17_{\pm 0.03}$ (0.00)	$67.18_{\pm 0.07}$ (0.00)	0.00
	GA	$3.86_{\pm 0.39}$ (3.86)	$71.40_{\pm 0.56}$ (17.66)	$65.27_{\pm 0.56}$ (10.90)	$45.06_{\pm 0.50}$ (22.12)	13.64
	GA+ours	$0.42_{\pm 0.15}$ (0.42)	$85.75_{\pm 0.52}$ (3.31)	$74.50_{\pm 0.26}$ (1.67)	$61.02_{\pm 0.19}$ (6.16)	**2.89**
	SalUn	$3.94_{\pm 0.24}$ (3.94)	$79.00_{\pm 0.73}$ (10.06)	$69.41_{\pm 0.81}$ (6.76)	$48.17_{\pm 0.21}$ (19.01)	9.94
	SalUn+ours	$0.56_{\pm 0.69}$ (0.56)	$84.87_{\pm 0.39}$ (4.19)	$74.06_{\pm 0.83}$ (2.11)	$60.30_{\pm 0.33}$ (6.88)	3.44
Class 9	Retrain	$0.00_{\pm 0.00}$ (0.00)	$88.72_{\pm 0.04}$ (0.00)	$76.30_{\pm 0.08}$ (0.00)	$68.69_{\pm 0.02}$ (0.00)	0.00
	GA	$5.30_{\pm 0.37}$ (5.30)	$75.14_{\pm 0.22}$ (13.58)	$66.19_{\pm 0.75}$ (10.11)	$45.06_{\pm 0.72}$ (23.63)	13.15
	GA+ours	$2.78_{\pm 0.23}$ (2.78)	$87.93_{\pm 0.43}$ (0.79)	$76.28_{\pm 0.51}$ (0.02)	$67.89_{\pm 0.38}$ (0.80)	**1.10**
	SalUn	$0.58_{\pm 0.08}$ (0.58)	$78.97_{\pm 0.42}$ (9.75)	$68.99_{\pm 0.28}$ (7.31)	$49.52_{\pm 1.01}$ (19.17)	9.20
	SalUn+ours	$4.69_{\pm 1.10}$ (4.69)	$81.47_{\pm 0.81}$ (7.25)	$71.43_{\pm 0.51}$ (4.87)	$56.46_{\pm 0.52}$ (12.23)	7.26

D.6 Multi-Class Unlearning

In this section, we evaluate GUARD's performance on the multi-classes unlearning tasks using the AllCNN model on the CIFAR-10 dataset. Specifically, the forget set D_f is constructed to include all samples from multiple target classes, while the retain set D_r comprises all the remaining data. Here, classes with index 2, 3, 4 were randomly selected as the target classes for forgetting. That is, we need to unlearn a D_f with 15000 samples at the same time.

Compared to single-class unlearning, removing multiple categories simultaneously is much more challenging, due to the large amount of forget data. As shown in Table A9, GU baselines exhibits a significantly higher risk of catastrophic forgetting on the retain data, as well as a larger extent of robustness degradation. Meanwhile, GUARD effectively reduces the accuracy of the target classes (2, 3, 4) to near-zero levels, while greatly preserving utility on the retain set D_r. Crucially, models unlearned using GUARD-enhanced baselines exhibit significant higher robust accuracy (close to Retrain oracle), indicating GUARD's superiority.

Table A9: Multi-class (class-2,3,4) unlearning results on CIFAR-10 with AllCNN.

	Methods	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\mathrm{RA}_r \uparrow$	Avg. Gap
	Original	$78.36_{\pm 0.16}$ (78.36)	$94.05_{\pm 0.08}$ (3.91)	$84.49_{\pm 0.10}$ (3.28)	$77.39_{\pm 0.21}$ (6.80)	23.09
	Retrain	$0.00_{\pm 0.00}$ (0.00)	$97.96_{\pm 0.02}$ (0.00)	$87.77_{\pm 0.04}$ (0.00)	$84.19_{\pm 0.13}$ (0.00)	0.00
	Muter	$9.89_{\pm 2.16}$ (9.89)	$93.16_{\pm 0.78}$ (4.80)	$83.77_{\pm 1.02}$ (4.00)	$66.23_{\pm 1.41}$ (17.96)	9.16
w/o D_r	GA	$2.02_{\pm 0.34}$ (2.02)	$83.34_{\pm 0.47}$ (14.62)	$75.02_{\pm 0.48}$ (12.75)	$54.78_{\pm 0.55}$ (29.41)	14.70
w/o D_r	**GA+ours**	$4.02_{\pm 0.19}$ (4.02)	$93.76_{\pm 0.33}$ (4.20)	$84.20_{\pm 0.41}$ (3.57)	$77.89_{\pm 0.55}$ (6.30)	4.52
w/o D_r	BS	$2.34_{\pm 0.45}$ (2.34)	$81.82_{\pm 0.27}$ (16.14)	$74.24_{\pm 0.64}$ (13.53)	$56.74_{\pm 0.71}$ (27.45)	14.87
w/o D_r	**BS+ours**	$3.86_{\pm 0.22}$ (3.86)	$93.37_{\pm 0.16}$ (4.59)	$85.43_{\pm 0.39}$ (2.34)	$79.26_{\pm 0.48}$ (4.93)	3.93
w/o D_r	SalUn	$0.00_{\pm 0.53}$ (0.00)	$84.50_{\pm 0.08}$ (13.46)	$75.68_{\pm 0.42}$ (12.09)	$58.38_{\pm 0.07}$ (25.81)	12.84
w/o D_r	**SalUn+ours**	$2.09_{\pm 0.29}$ (2.09)	$93.08_{\pm 0.14}$ (4.88)	$84.77_{\pm 0.35}$ (3.00)	$78.71_{\pm 0.52}$ (5.48)	3.86
w/ D_r	FT	$0.37_{\pm 0.61}$ (0.37)	$97.02_{\pm 0.23}$ (0.94)	$86.51_{\pm 0.46}$ (1.26)	$53.67_{\pm 0.82}$ (30.52)	8.27
w/ D_r	**FT+ours**	$5.11_{\pm 0.06}$ (5.11)	$95.85_{\pm 0.10}$ (2.11)	$86.04_{\pm 0.06}$ (1.73)	$73.58_{\pm 0.08}$ (10.61)	4.89
w/ D_r	UNSIR	$0.81_{\pm 0.13}$ (0.81)	$96.68_{\pm 0.19}$ (1.28)	$87.29_{\pm 0.25}$ (0.48)	$65.01_{\pm 0.21}$ (19.18)	5.44
w/ D_r	**UNSIR+ours**	$1.71_{\pm 0.27}$ (1.71)	$96.70_{\pm 0.60}$ (1.26)	$86.59_{\pm 0.22}$ (1.18)	$74.74_{\pm 0.19}$ (9.45)	**3.40**

D.7 Sequential Unlearning Results

We evaluate sequential unlearning of five classes (indices 3, 4, 5, 7, and 8) on AllCNN/CIFAR-10 with GA+GUARD. As shown in Table A10, GUARD effectively fulfills successive forgetting requests while maintaining low forget accuracy. RA remains stable within 75.12–82.30%, showing no cumulative degradation in adversarial robustness. These results demonstrate GUARD's effectiveness in preserving robust decision boundaries under sequential unlearning.

Table A10: Sequential unlearning on AllCNN, CIFAR-10, GA+GUARD.

Class	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$RA_r \uparrow$
3	0.76	92.60	81.91	75.12
3+4	2.60	95.18	85.25	80.00
3+4+5	2.92	96.50	87.21	82.30
3+4+5+7	2.16	95.76	87.02	80.38
3+4+5+7+8	1.40	95.82	87.88	81.24

D.8 Evaluation through Membership Inference Attacks (MIA)

To quantitatively assess the privacy protection capabilities of our proposed GUARD, we employ Membership Inference Attacks (MIA). This approach is grounded in the premise that an adversary can infer whether a specific sample was part of the model's training set by analyzing the target model's output posterior probabilities.

In our setup, we define Member Data as samples from the forget set $\mathcal{D}_f$ and Non-Member Data as unseen samples from the test forget set (test data belonging to the target forget class). We ensure these two groups have the same size. They are then merged to form the MIA training set used to train the attack mode. Then, we utilize the original model's Softmax probability vectors from the MIA training set to train a binary Support Vector Classifier (SVC), enabling the attacker to learn the high-confidence patterns typical of training members.

We apply this trained attack model to the unlearned model's outputs with respect to the forget set $\mathcal{D}_f$. We evaluate the unlearning performance through the Attack Success Rate (ASR) of the MIA. A high ASR confirms that the attack model is functional and capable of correctly identifying unseen data patterns; a low ASR on these forgotten samples indicates that the model's memory has been effectively erased. Ideally, the ASR on the unlearned model is expected to approximate that on the Retrain oracle, achieving near-0 level.

We conduct MIA evaluation on the class unlearning task on AllCNN and CIFAR-10, with class-3 being the target class. As shown in Table A11, the ASR in both baselines and GUARD-enhanced baselines is significantly lower than that of the original model and remarkably close to the Retrain oracle. These results demonstrate that GUARD effectively erases membership information, safeguarding the privacy of the forget data.

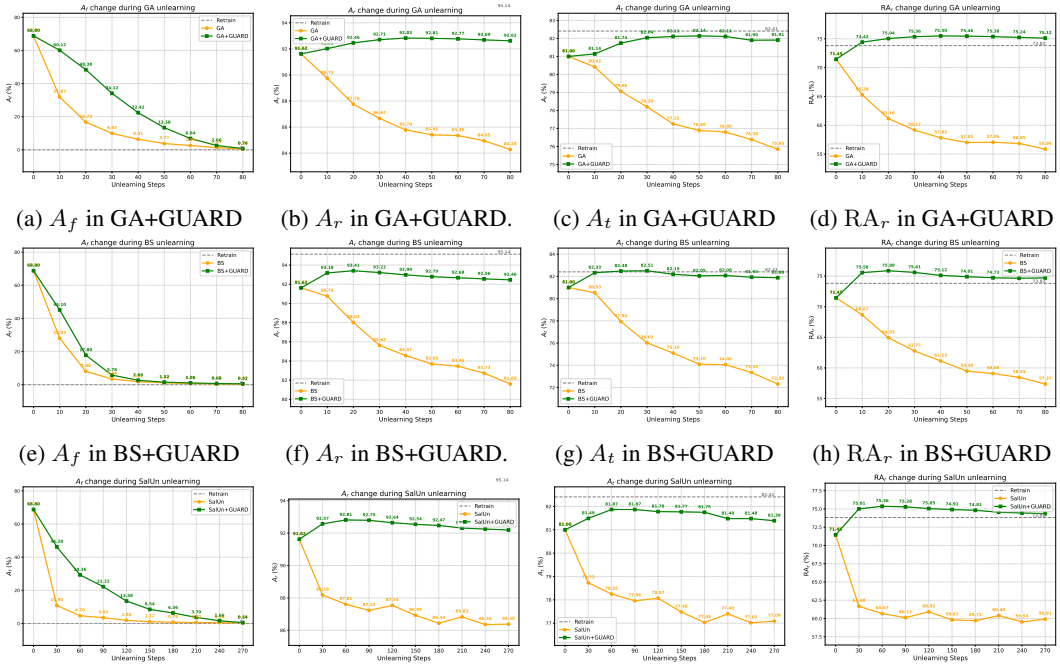
Table A11: MIA attack success rate (%) of baselines across 4 settings.

Model	AllCNN		ResNet-18	
Dataset	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
Original	$73.00_{\pm 0.20}$ (73.00)	$100.0_{\pm 0.00}$ (100.0)	$100.0_{\pm 0.00}$ (100.0)	$100.0_{\pm 0.00}$ (100.0)
Retrain	$0.00_{\pm 0.00}$ (0.00)	$0.00_{\pm 0.00}$ (0.00)	$0.00_{\pm 0.00}$ (0.00)	$0.00_{\pm 0.00}$ (0.00)
GA	$5.00_{\pm 0.10}$ (5.00)	$0.00_{\pm 0.00}$ (0.00)	$0.00_{\pm 0.00}$ (0.00)	$0.00_{\pm 0.00}$ (0.00)
GA+ours	$6.50_{\pm 0.05}$ (6.50)	$0.50_{\pm 0.05}$ (0.50)	$0.00_{\pm 0.00}$ (0.00)	$0.00_{\pm 0.00}$ (0.00)
BS	$0.00_{\pm 0.00}$ (0.00)	$3.20_{\pm 0.20}$ (3.20)	$2.00_{\pm 0.05}$ (2.00)	$1.50_{\pm 0.05}$ (1.50)
BS+ours	$0.50_{\pm 0.00}$ (0.50)	$0.00_{\pm 0.00}$ (0.00)	$0.00_{\pm 0.00}$ (0.00)	$0.00_{\pm 0.00}$ (0.00)
SalUn	$0.00_{\pm 0.00}$ (0.00)	$0.00_{\pm 0.00}$ (0.00)	$0.05_{\pm 0.00}$ (0.05)	$0.00_{\pm 0.00}$ (0.00)
SalUn+ours	$0.85_{\pm 0.50}$ (0.85)	$0.00_{\pm 0.00}$ (0.00)	$0.00_{\pm 0.00}$ (0.00)	$1.00_{\pm 0.05}$ (1.00)

D.9 Evolution of Metrics During Unlearning

In this section, we visualize the change of the 4 accuracy metrics throughout the class unlearning process, with AllCNN on CIFAR-10, in terms of 3 baselines (GA, BS, SalUn) and the corresponding GUARD-enhanced version. These trajectories demonstrate how the unlearning algorithm influences model performance as the unlearning intensity increases.

We conduct a direct comparison between the baseline (orange line), its GUARD-enhanced version (green line), and the Retrain oracle (gray line). As illustrated in Fig. A6, we observe: (i) Regarding A_f: The GUARD curve exhibits a smoother trajectory, indicating a more stable unlearning process for D_f. This suggests that GUARD applies a more consistent and moderate intensity of modification to the decision boundaries.(ii) Regarding RAr: The GUARD curve is significantly higher than the baseline and closely matches—or even surpasses—the Retrain oracle, showing negligible degradation compared to the original model. This demonstrates the superior efficacy of GUARD in Robustness Preservation.(iii) Regarding A_r and A_t: GUARD mirrors the trends observed in RAr, maintaining performance levels comparable to the Retrain oracle. This highlights its exceptional capability in preserving model Utility.In summary, these experiments show the superiority of GUARD across multiple performance dimensions.



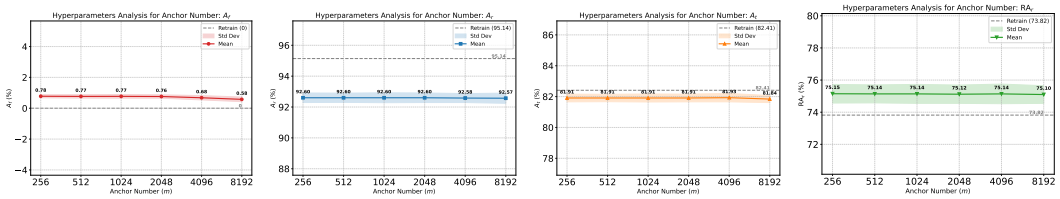
(a) A_f in GA+GUARD (b) A_r in GA+GUARD. (c) A_t in GA+GUARD (d) RA_r in GA+GUARD

(e) A_f in BS+GUARD (f) A_r in BS+GUARD. (g) A_t in BS+GUARD (h) RA_r in BS+GUARD

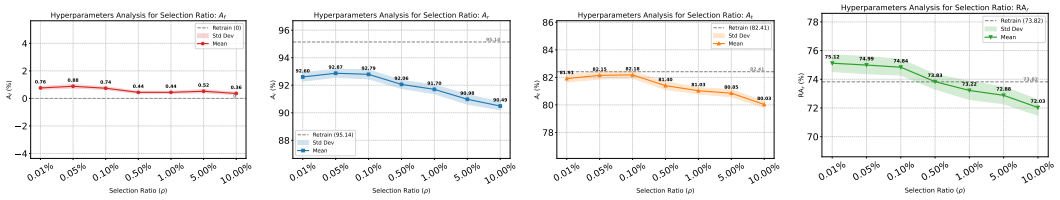
(i) A_f in SalUn+GUARD (j) A_r in SalUn+GUARD. (k) A_t in SalUn+GUARD (l) RA_r in SalUn+GUARD

Figure A6: Evolution of metrics during 3 unlearning baselines on AllCNN, CIFAR-10.

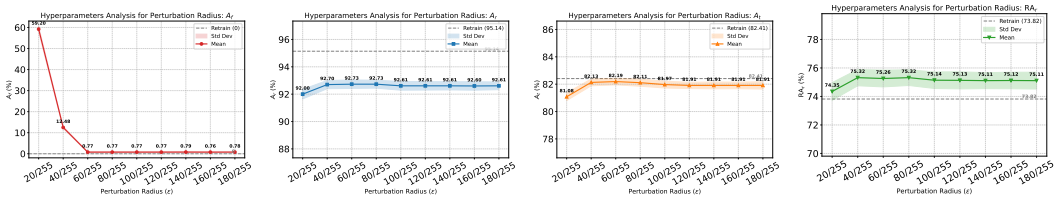
D.10 Extended Results for Hyperparameter Analysis



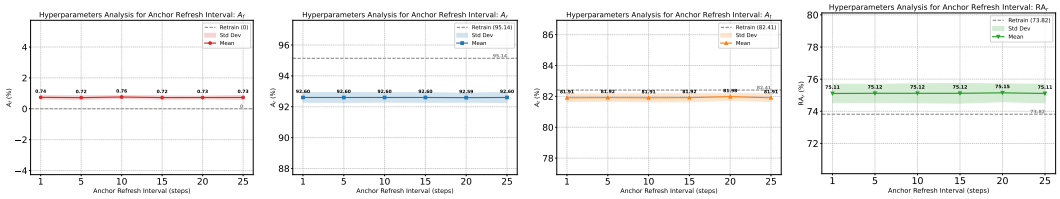
(a) A_f vs. anchor number. (b) A_r vs. anchor number. (c) A_t vs. anchor number. (d) RA_r vs. anchor number.



(e) A_f vs. selection ratio (f) A_r vs. selection ratio (g) A_t vs. selection ratio (h) RA_r vs. selection ratio



(i) A_f vs. perturbation radius ϵ (j) A_r vs. perturbation radius ϵ (k) A_t vs. perturbation radius ϵ (l) RA_r vs. perturbation radius ϵ.



(m) A_f vs refresh intervals (n) A_r vs. refresh intervals (o) A_t vs. refresh intervals (p) RA_r vs refresh intervals

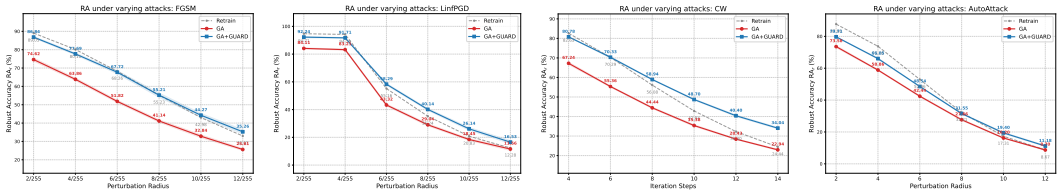
Figure A7: Hyperparameter analysis with GA on AllCNN, CIFAR-10. We draw the change curves of four metrics (A_f, A_r, A_t, RA_r) with respect to four key hyperparameters (Anchor Number, Selection Ratio, Perturbation Radius, Refresh Intervals).

In this section, we present the comprehensive hyperparameter sensitivity analysis. We report the trends of all four metrics (A_f, A_r, A_t, RA_r) as we vary the four principal hyperparameters: the anchor number, selection ratio (ρ), perturbation radius for simulating $\mathcal{D}_v$ (ϵ), and anchor refresh interval. These results provide a holistic view of how each hyperparameter influences the trade-off between unlearning efficacy, model utility, and robustness.

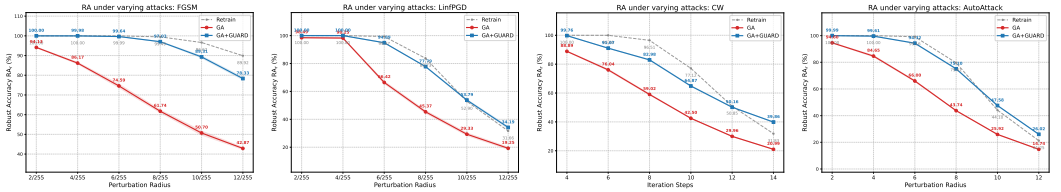
As illustrated in Fig. A7a, A7b, A7c, A7d and Fig. A7m, A7n, A7o, A7p, GUARD exhibits steady performance to variations in both the anchor number and the anchor refresh interval, consistently showing performance comparable to the Retrain baseline across all four metrics. This suggests that, in practice, a smaller anchor set (e.g., 32) and a lower refresh frequency (e.g. 10 steps) can be adopted to significantly enhance computational efficiency without compromising performance. However, Fig. A7i, A7j, A7k, A7l demonstrate that while GUARD maintains stable in most settings, utilizing extremely small adversarial perturbations (specifically, less than $60/255$) when generating simulated D_v can significantly degrade unlearning effectiveness. Finally, regarding the selection ratio

ρ, Fig. A7e, A7f, A7g, A7h show a consistent improvement across all metrics as ρ decreases. This trend validates the efficacy of our Importance-Guided Parameter Selection strategy.

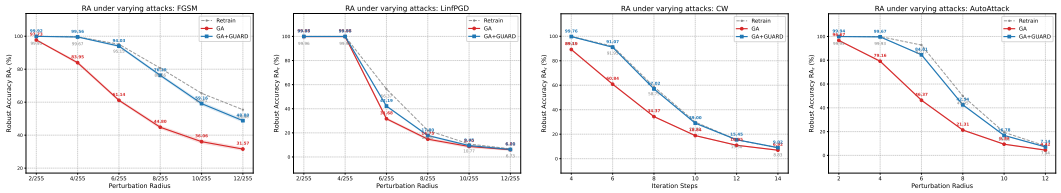
D.11 Extended Results under Varying Attacks



(a) FGSM on AllCNN with CIFAR-10 (b) ℓ_{inf}-PGD on AllCNN with CIFAR-10 (c) CW on AllCNN with CIFAR-10 (d) AutoAttack on AllCNN with CIFAR-10



(e) FGSM on ResNet-18 with CIFAR-10 (f) ℓ_{inf}-PGD on ResNet-18 with CIFAR-10 (g) CW on ResNet-18 with CIFAR-10 (h) AutoAttack on ResNet-18 with CIFAR-10



(i) FGSM on AllCNN with CIFAR-100 (j) ℓ_{inf}-PGD on AllCNN with CIFAR-100 (k) CW on AllCNN with CIFAR-100 (l) AutoAttack on ResNet-18 with CIFAR-100

Figure A8: Robust Accuracy RA_r change under 4 attacks and varying strengths using GA as baseline on 2 datasets and 2 architectures.

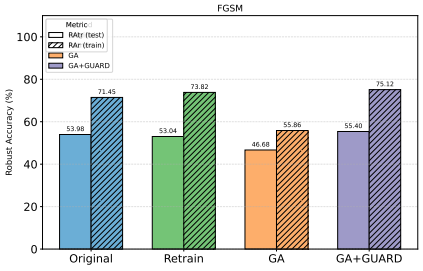
The detailed settings for these adversarial attacks are detailed here. For ℓ_{inf}-PGD attack, we set the number of iterations to 20 and the perturbation size α at each iteration to $\epsilon/10$, where ϵ denotes the overall perturbation budget. For the CW attack, we set the confidence margin κ to 0, the trade-off coefficient c between attack success and perturbation size to 1×10^{-3}, the learning rate to 0.01, the number of binary search steps for tuning c to 4, and the maximum number of optimization iterations to 10. For AutoAttack, we used the standard evaluation mode, which consists of APGD-CE, APGD-T, FAB-T, and Square Attack, with 100 iterations for APGD/FAB-based attacks, one restart, nine target classes for targeted attacks, and 5000 queries for Square Attack.

Complementing the analysis in the main text, we here present the full experimental results across 2 datasets and 2 model architectures. As illustrated in Fig. A8, GUARD consistently demonstrates superior capability in preserving adversarial robustness compared to the GA baseline, achieving a performance close to the Retrain oracle. These extended results further validate the generalizability of our proposed method across different settings.

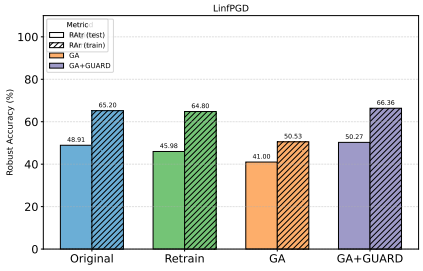
It is important to note that GUARD is designed to preserve the intrinsic adversarial robustness of the original model, rather than introducing additional resilience. Since all evaluations here are based on the Original and Retrain model adversarially trained using FGSM with $\epsilon = 0.02$, a general decline in robustness is expected for all models when subjected to attack strengths significantly exceeding this training threshold. However, GUARD maintains significantly higher robustness compared to the GA baseline, and achieves performance closely approximating that of the Retrain model. This

confirms that although GUARD is bound by the original model's capacity, it effectively prevents the catastrophic forgetting of robustness observed in standard unlearning methods.

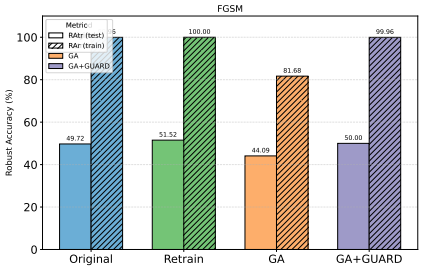
D.12 Extended Results for Adversarial Attacks on Unseen Data



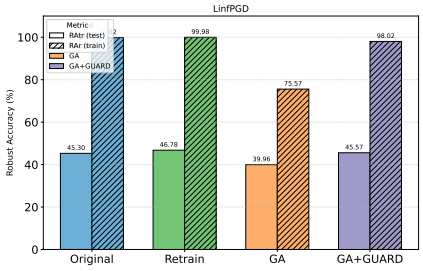
(a) FGSM, AllCNN, CIFAR-10



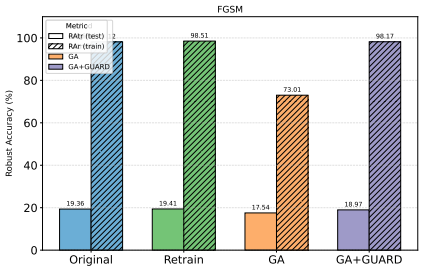
(b) ℓ_∞-PGD, AllCNN, CIFAR-10



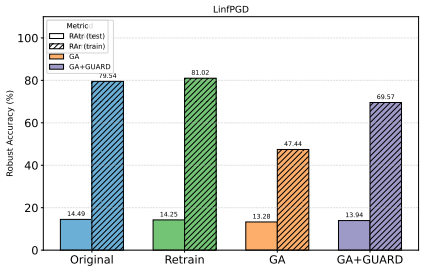
(c) FGSM, ResNet-18, CIFAR-10



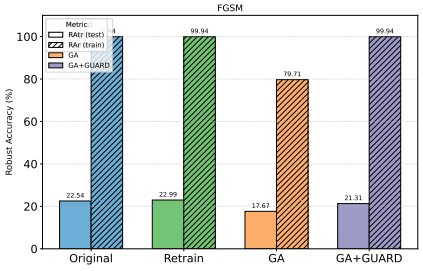
(d) ℓ_∞-PGD, ResNet-18, CIFAR-10



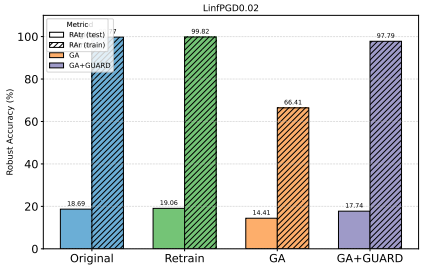
(e) FGSM, AllCNN, CIFAR-100



(f) ℓ_∞-PGD, AllCNN, CIFAR-100



(g) FGSM, ResNet-18, CIFAR-100



(h) ℓ_∞-PGD, ResNet-18, CIFAR-100

Figure A9: Comparison between robust accuracy on unseen test retain data RA_{tr} and robust accuracy on the training retain set RA_r on CIFAR-10 under GA-based attacks.

To further evaluate the adversarial robustness of different models beyond the training data, we conduct additional attacks on unseen test samples. Both FGSM and ℓ_∞-PGD use a perturbation budget of $\epsilon = 0.02$, and ℓ_∞-PGD is run for 20 iterations. This evaluation allows us to examine whether the robustness preserved by GUARD can generalize to previously unseen data rather than merely fitting

the adversarial patterns observed during training. Our goal is to achieve robustness comparable to the retrain oracle, which serves as an upper-bound reference.

As shown in Figure A9, we observe that the gap between RA_{tr} and RA_r is relatively small on CIFAR-10, while it becomes much larger on CIFAR-100. This is expected, since CIFAR-100 contains more classes and fewer samples per class, making robust generalization to unseen retain data more challenging. Therefore, our main focus is not the absolute gap between RA_{tr} and RA_r, but whether GUARD can approach the Retrain oracle under the same evaluation setting. Compared with naive GA, GUARD consistently achieves better robust accuracy close to the Retrain oracle, showing that it better preserves robustness on unseen retain data than the GU baseline.

D.13 Extended Results for PGD/TRADES-trained Base Models

We adopt FGSM-based adversarial training (AT) in our main experiments due to its computational efficiency and practical applicability. Compared with stronger AT methods, FGSM-trained models may exhibit narrower robustness margins, making retained samples more susceptible to robustness degradation during machine unlearning (MU). This provides a sensitive setting for evaluating the importance of robustness-preserving unlearning.

To address concerns about potential catastrophic overfitting associated with FGSM-based AT, we evaluate GUARD against stronger adversarial attacks, including PGD and AutoAttack. The results consistently demonstrate its effectiveness under these attacks, indicating that the observed robustness improvements are not merely artifacts of FGSM-based evaluation.

We further extend our experiments to four additional base models trained with PGD-10 and TRADES, covering both AllCNN and ResNet-18, using GA as the baseline. As shown in Table 12, GUARD consistently improves robust accuracy (RA) by 4.78–12.34 percentage points over vanilla GA across all four settings, while further reducing forgetting accuracy (A_f). Notably, GUARD achieves RA improvements of 12.34 and 12.05 percentage points on AllCNN-PGD10 and ResNet-18-PGD10, respectively. Although slight clean-accuracy degradation occurs on ResNet-18-PGD10, the substantial robustness gain demonstrates the effectiveness of GUARD in mitigating robustness degradation induced by MU. These results confirm that GUARD is compatible with different AT paradigms and generalizes beyond FGSM-trained models to stronger PGD- and TRADES-trained models.

Table A12: Results on PGD-10/TRADES-trained base models.

AT	Method	AllCNN				ResNet-18			
		$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\mathrm{RA}_r \uparrow$	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\mathrm{RA}_r \uparrow$
PGD-10	Base	55.32	83.35	80.10	53.42	67.78	90.59	85.93	75.24
	GA	1.48	76.00	73.52	37.78	2.24	86.67	79.61	48.10
	GA+GUARD	0.36	79.70	76.70	**50.12**	1.84	84.47	77.70	**60.15**
TRADES	Base	61.04	77.12	75.57	67.53	72.24	86.20	82.47	77.03
	GA	2.80	71.92	70.84	59.45	3.50	82.10	78.50	67.26
	GA+GUARD	1.36	78.96	77.46	**68.84**	2.52	83.68	79.92	**72.04**

D.14 Comparison with Post-Unlearning Adversarial Training

Applying adversarial training (AT) after unlearning is a natural alternative for restoring model robustness. To investigate this approach, we implement a post-unlearning AT baseline on AllCNN/CIFAR-100 using GA as the unlearning method. Specifically, we apply FGSM-based AT with a perturbation budget of $\epsilon = 0.02$ to the unlearned model, using retain data for adversarial optimization. We examine different learning rates and training epochs, and compare the resulting forgetting performance, model utility, robust accuracy, and computational cost with GA+GUARD.

Table A13: **Comparison with post-unlearning adversarial training.** For GA+AT, runtime is reported as unlearning time + additional AT time; for GA+GUARD, the end-to-end runtime is reported.

Method	LR	Epochs	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\mathrm{RA}_r \uparrow$	Runtime (s)
GA+AT	0.01	10	100.0	99.96	54.42	99.91	$2.6 + 100.5$
GA+AT	0.01	30	100.0	99.98	54.60	99.93	$2.6 + 293.6$
GA+AT	0.1	30	10.80	99.95	52.16	95.72	$2.6 + 301.1$
GA+GUARD	–	–	0.00	99.96	53.30	**98.17**	**12.8**

Table A13 presents three representative configurations of post-unlearning AT. With a relatively small learning rate of 0.01, AT achieves high robust accuracy (99.91–99.93%) but completely restores the forgotten knowledge ($A_f = 100\%$), even when the training duration is extended from 10 to 30 epochs. Increasing the learning rate to 0.1 and training for 30 epochs substantially reduces A_f to 10.80%, while maintaining an RA of 95.72%. However, this improvement requires 301.1 seconds of additional AT, beyond the 2.6-second unlearning stage. In contrast, GA+GUARD achieves stronger forgetting ($A_f = 0.00\%$) and higher RA_r (98.17%) in only 12.8 seconds, reducing the total runtime by approximately 23.7×.

These results reveal a fundamental challenge in post-unlearning AT: restoring robustness does not necessarily preserve the forgetting objective. One possible explanation is that small-learning-rate AT primarily adjusts local decision boundaries using retain data. Due to shared representations between forget and retain classes, such updates may inadvertently reactivate previously forgotten features. A larger learning rate allows more substantial changes to the decision boundaries, potentially improving the forgetting–robustness trade-off, but at the cost of extensive adversarial optimization. In contrast, GUARD explicitly accounts for robustness preservation during unlearning, avoiding the need for a separate robustness-recovery stage.

Table A14: **Effect of retain-data size on post-unlearning AT.** The number of training epochs is adjusted to approximately match the total optimization steps across different retain-set sizes. Total runtime includes the initial 2.6-second GA unlearning stage for post-AT configurations.

Samples/class	$\lvert\mathcal{D}_r\rvert$	Epochs	$A_f \downarrow$	$A_r \uparrow$	$A_t \uparrow$	$\mathrm{RA}_r \uparrow$	AT Time (s)	Total (s)
1	99	11610	99.80	99.95	52.75	96.10	191.0	193.6
5	495	2903	100.0	99.96	54.07	97.73	217.7	220.3
10	990	1452	99.80	99.95	53.32	95.17	219.9	222.5
25	2475	596	93.20	99.41	50.79	67.25	223.0	225.6
50	4950	298	87.60	99.05	51.24	62.70	223.2	225.8
100	9900	150	76.20	99.78	53.63	77.01	223.7	226.3
250	24750	60	54.60	99.65	53.33	79.03	224.4	227.0
500	49500	30	17.80	99.95	52.53	94.91	224.7	227.3
GA+GUARD (anchor size = 32)			0.00	99.96	53.30	**98.17**	–	**12.8**

We further investigate the sensitivity of post-unlearning AT to the amount of available retain data. To ensure a comparable optimization budget, we adjust the number of training epochs to maintain approximately the same total number of gradient-update steps across different retain-set sizes. As shown in Table A14, smaller retain sets lead to severe forgetting recovery, with A_f reaching 99.80–100.0% when using only 99–990 retain samples. Although these configurations achieve relatively high RA_r, they largely undermine the original unlearning objective.

As the retain-set size increases, forgetting recovery is generally alleviated, whereas robust accuracy exhibits considerable fluctuations. In particular, intermediate retain-set sizes (2,475–4,950 samples) result in substantial robustness degradation, with RA_r dropping to 62.70–67.25%. Even with 49,500 retain samples, post-unlearning AT still recovers 17.80% of the forgotten knowledge and achieves an RA_r of only 94.91%, requiring 224.7 seconds of additional AT. In contrast, GA+GUARD requires no retain data and achieves complete forgetting ($A_f = 0.00\%$) with higher RA (98.17%) in only 12.8 seconds, using a compact vulnerable set $\mathcal{D}_v$ constructed from forget-data-derived anchors.

These findings highlight three advantages of GUARD over post-unlearning AT. First, post-unlearning AT depends on access to retain data, whereas GUARD supports retain-free unlearning by constructing a compact simulated vulnerable set. Second, post-unlearning AT introduces substantial computational overhead, even with the relatively efficient FGSM-based AT, while GUARD incorporates robustness preservation directly into the unlearning process. Third, post-unlearning AT optimizes robustness without explicitly constraining the forgetting objective, potentially recovering forgotten knowledge through shared representations between forget and retain data. By contrast, GUARD directly restricts robustness-degrading unlearning updates, enabling a more favorable balance between forgetting effectiveness, robustness preservation, and computational efficiency.

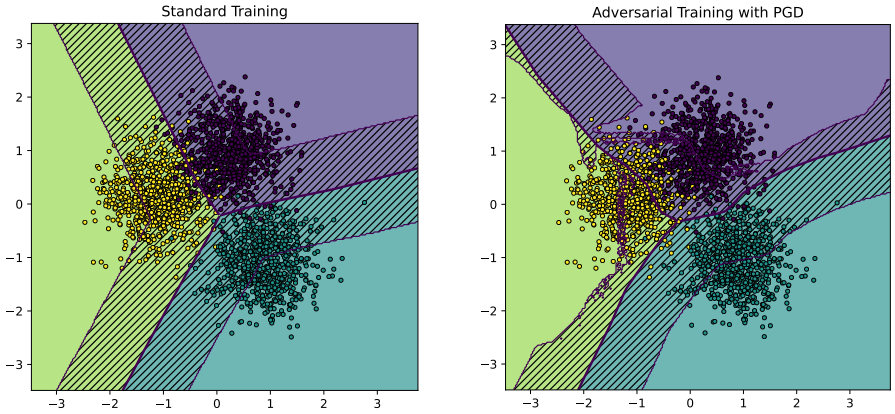
E Visualization

E.1 A Geometric Perspective on Robustness Degradation: A Toy Example

To intuitively demonstrate how GU compromises the adversarial robustness of the retain data, we conducted a geometric visualization using a toy example in Fig. A10. We constructed a synthetic 2D dataset consisting of three Gaussian clusters (Classes 0, 1, and 2), where Class 1 is set as the forget set and the others form the retain set. Using this data, we trained a Multi-Layer Perceptron (MLP) via both standard and adversarial training (with PGD). To visualize the non-robust zones (represented by hatched regions), we performed a high-resolution grid scan where each coordinate is marked as non-robust if a ϵ-radius PGD attack can successfully flip its predicted label. Within these regions, although data points may be correctly classified, they are highly susceptible to adversarial perturbations. A comparison between the standard trained model in Fig. A10a and the Adversarially Trained (AT) model in Fig. A10b reveals that AT significantly compresses the width of the non-robust zones along the classification bouandaries, forming a robust decision boundary, thereby ensuring the model with strong resistance against attacks.

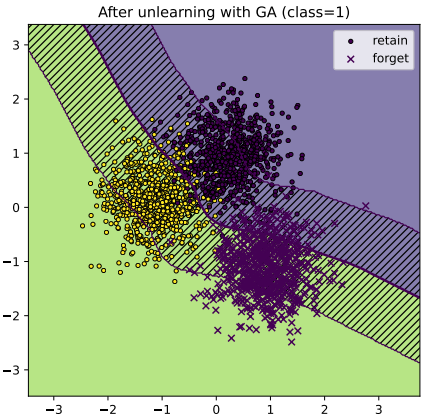
On the AT model, we applied naive Gradient Ascent (GA), yielding the decision boundaries depicted in Fig. A10c. This visualization reveals a catastrophic degradation of the robust boundaries associated with the retain data. Although the model successfully induces misclassification on Class-1 samples (forgetting the target knowledge), this drastic shift in decision boundaries causes the previously narrow non-robust zones to expand. Although the clean accuracy of the retain set remains largely unaffected, these samples have mostly fallen into the non-robust zones, rendering them highly vulnerable to adversarial attacks. This geometric phenomenon confirms that, without geometric constraints on robust decision boundaries, the unlearning objective is often achieved at the expense of the retain-data robustness.

Similarly, we applied the GUARD-enhanced GA to the AT model. Unlike naive GA, GUARD imposes explicit constraints on the movement of decision boundaries during the unlearning process via importance-guided parameter selection and gradient projection. The visualization results in Fig. A10d demonstrate that the unlearned model not only achieves effective unlearning but, crucially, preserves the robust decision boundaries originally established by adversarial training. By maintaining the narrow non-robust zones, GUARD ensures that the majority of the retain data remains within adversarially safe regions. This confirms that our method effectively decouples the conflict between the unlearning objective and robustness preservation, ensuring the geometric structural of the model.

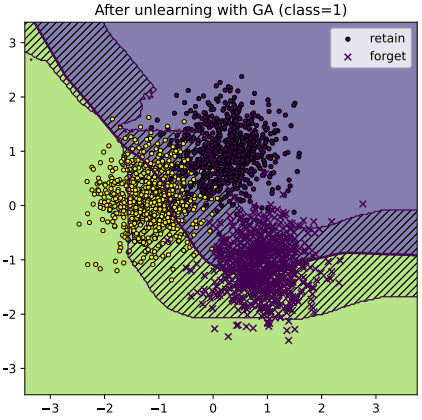


(a) Standard Training Model

(b) Adversarial Training Model



(c) Unlearned Model After GA



(d) Unlearned Model After GA+GUARD

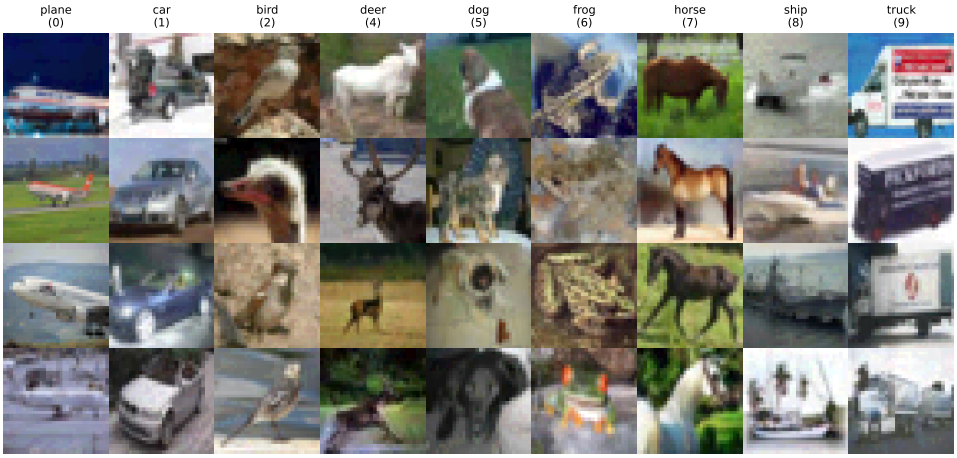
Figure A10: Visualisatoin of a toy example. Dots are synthetic data points, the hatched regions represent the non-robust zones, and the thick solid lines are the clean-accuracy decision boundary.

E.2 Constructed Adversarially Vulnerable Dataset

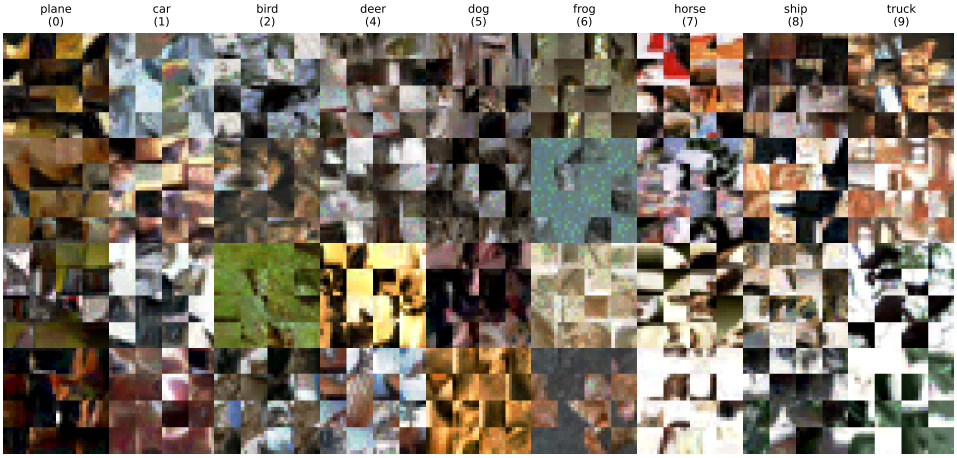
This section visualizes samples from the constructed Adversarially Vulnerable Dataset ($\mathcal{D}_v$) to provide an intuitive understanding of the generated features.

Fig. A11a displays the $\mathcal{D}_v$ constructed directly from the retain data $\mathcal{D}_r$. While the adversarial perturbations remain visually imperceptible to the human eye, they significantly degrade the model's discriminative capability, confirming that these samples are located at the decision boundaries.

In contrast, Fig. A11b presents the simulated $\mathcal{D}_v$ derived from the forget data $\mathcal{D}_f$. Subjected to strong data augmentations (e.g., random patching), these images have lost the interpretable semantic information associated with the original $\mathcal{D}_f$. Furthermore, the use of a relatively large perturbation radius allows these samples to traverse the feature space and successfully anchor to the proximity of the retain classes' decision boundaries.



(a) Adversarially vulnerable dataset constructed from $\mathcal{D}_r$, when unlearning class-3 with AllCNN on CIFAR-10.



(b) Simulated adversarially vulnerable set constructed from $\mathcal{D}_f$, when unlearning class-3 on AllCNN, CIFAR-10.

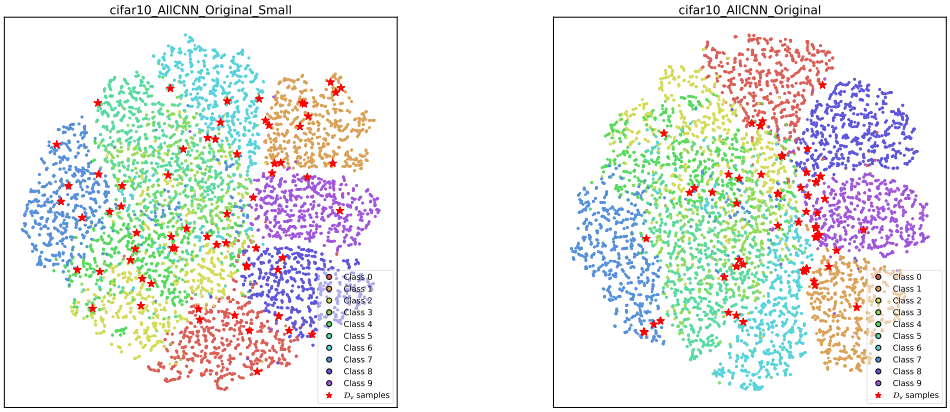
Figure A11: Visualisation of the constructed adversarially vulnerable set.

E.3 T-SNE Map

To verify whether the adversarially vulnerable set $\mathcal{D}_v$ effectively captures the boundary features critical for robustness, we use t-SNE [36] to visualize the feature space distributions of the normal retain samples $\mathcal{D}_r$ and the constructed vulnerable set $\mathcal{D}_v$. As shown in Fig. A12a, analyzing the real $\mathcal{D}_v$ constructed from $\mathcal{D}_r$, we observe that the majority of samples are located at the periphery of class clusters rather than near high-density centroids. This geometric distribution confirms our hypothesis that $\mathcal{D}_v$ successfully localizes boundary-proximal regions, comprising samples with minimal adversarial margins that are most susceptible to attacks. Furthermore, as shown in Fig. A12b, the simulated $\mathcal{D}_v$ (derived from $\mathcal{D}_f$) exhibits a distribution pattern remarkably similar to that of the real $\mathcal{D}_v$. Despite the distribution shift in the source data, the simulated samples are effectively guided toward the boundary subspaces of the retain classes. This phenomenon validates the effectiveness of our simulation strategy in retain-free scenarios, demonstrating that $\mathcal{D}_f$ can be leveraged to reproduce the adversarial vulnerability features of the target data. In summary, the visualization attests to the geometric validity of $\mathcal{D}_v$ as an efficient and lightweight proxy for anchoring and preserving adversarial robustness during machine unlearning.

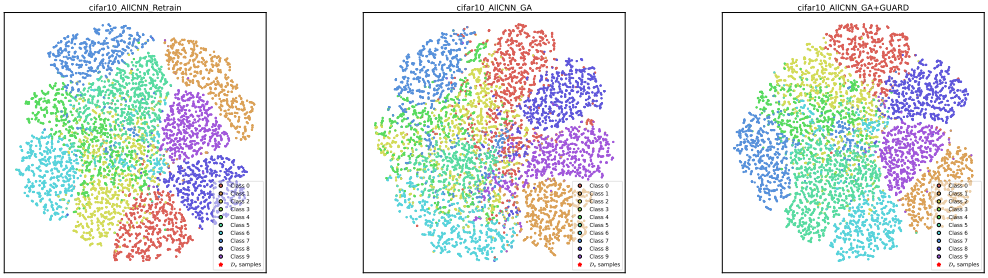
To further investigate the impact of the GU process on discriminative structure and robust representations, we employ t-SNE to visualize the feature spaces of three models: Retrain-from-scratch (the gold standard), the Baseline (GA as an example), and GUARD-enhanced method (Baseline+GUARD). This visualization aims to reveal two critical aspects: whether the unlearning process effectively

eliminates the distinct discriminative structure of the forget class, and whether it compromises the intra-class compactness or inter-class margins of the remaining classes, which actually indicates a degradation in robustness. Comparative results in Fig. A13 show that the Baseline often introduces significant representation drift and boundary perturbations. In contrast, applying our framework achieves the unlearning objective while maintaining cluster structures and separation margins comparable to the Retrain-from-scratch model. This suggests that our approach restricts parameter updates to localized boundary reshaping, thereby better preserving the robust discriminative geometry.



(a) $\mathcal{D}_v$ constructed using $\mathcal{D}_r$. (b) $\mathcal{D}_v$ simulated using $\mathcal{D}_f$.

Figure A12: t-SNE visualization of the constructed adversarially vulnerable set. (a) shows the set constructed from the retain data, while (b) shows our simulated set derived from the forget data.



(a) Retrain-from-scratch model. (b) Unlearned model using GA. (c) Unlearned by GA+GUARD.

Figure A13: t-SNE visualization of the result models after unlearning class-3 on AllCNN, CIFAR-10.

E.4 Limitations.

In the retain-free setting, constructing the proxy adversarially vulnerable dataset ($\mathcal{D}_v$) solely through semantic augmentations of the forget set may not fully recover the complex decision boundaries of unseen retain classes, especially when the forget and retain classes are completely disjoint. As a result, the constructed proxy may introduce a certain degree of approximation error. Although our empirical evaluation suggests that this proxy is effective and reliable in practice, we do not provide a formal bound on such errors and do not evaluate under extremely complex scenarios, and thus this construction lacks a rigorous theoretical guarantee. Compared with data generation methods such as GANs, our simulation strategy trades some fidelity for improved efficiency and scalability.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction(section 1) section clearly detailed our contribution and insights.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We stated the limitation of our work in Appendix E.4.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provided our main theoretical results in Section 6 of the main body, and we detailed all the derivatives and proofs in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We detailed the settings and hyperparameter required to reproduce our results in Section 7.1 in the main body, as well as in Appendix D.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We uploaded all the experiment code, and the dataset used are open access.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We detailed the settings and hyperparameter required to reproduce our results in Section 7.1 in the main body, as well as in Appendix D.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We stated that we run 5 times with random seed, and report the stdand deviation in Section 7.1 of the main body.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided the hardware and software information in Section 7.1 of the main body, and we reported the time cost (seconds) of running our method, in Section 7.6 and Appendix D.4. We also gave space and computational complexity analysis in Appendix C.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We strictly follow the code of ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We stated the potential influence to future study in Section 8 of the main body.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release any dataset or models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the codes, data and models used in the paper are properly credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide a README.md document along with the codes we uploaded.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.