

# BEYOND SHARP MINIMA: ROBUST LLM UNLEARNING VIA FEEDBACK-GUIDED MULTI-POINT OPTIMIZATION

**Wenhan Wu\***

Department of Statistics and Data Science  
Northwestern University  
Evanston, IL 60208, USA  
wuwenhan564@gmail.com

**Zheyuan Liu**

Department of Computer Science and Engineering  
University of Notre Dame  
Notre Dame, IN 46556, USA  
zliu29@nd.edu

**Chongyang Gao**

Department of Computer Science  
Northwestern University  
Evanston, IL 60208, USA  
Chongyanggao2026@u.northwestern.edu

**Ren Wang**

Department of Electrical and Computer Engineering  
Illinois Institute of Technology  
Chicago, IL 60616-3793, USA  
rwang74@iit.edu

**Kaize Ding**

Department of Statistics and Data Science  
Northwestern University  
Evanston, IL 60208, USA  
kaize.ding@northwestern.edu

## ABSTRACT

Current LLM unlearning methods face a critical security vulnerability that undermines their fundamental purpose: while they appear to successfully remove sensitive or harmful knowledge, this “forgotten” information remains precariously recoverable through relearning attacks. We identify that the root cause is that conventional methods optimizing the forgetting loss at individual data points will drive model parameters toward sharp minima in the loss landscape. In these unstable regions, even minimal parameter perturbations can drastically alter the model’s behavior. Consequently, relearning attacks exploit this vulnerability by using just a few fine-tuning samples to navigate the steep gradients surrounding these unstable regions, thereby rapidly recovering knowledge that was supposedly erased. This exposes a critical robustness gap between apparent unlearning and actual knowledge removal. To address this issue, we propose StableUN, a bi-level feedback-guided optimization framework that explicitly seeks more stable parameter regions via neighborhood-aware optimization. It integrates forgetting feedback, which uses adversarial perturbations to probe parameter neighborhoods, with remembering feedback to preserve model utility, aligning the two objectives through gradient projection. Experiments on WMDP and MUSE benchmarks demonstrate that our method is significantly more robust against both relearning and jailbreaking attacks while maintaining competitive utility performance.

## 1 INTRODUCTION

Recently, Large Language Models (LLMs) have quickly become the cornerstone of a wide array of modern systems, powering everything from task-oriented assistants to domain-focused knowledge engines (Dong et al., 2023; Kazemitaab et al., 2024; Guo et al., 2024; Qiao et al., 2024). However, their proliferation raises pressing concerns about privacy, safety, and trustworthiness. This is

---

\*Intern at Northwestern University

because LLMs’ training corpora often contain sensitive, biased, or even unlawful content that poses significant risks when deployed in real-world applications. In addition, data-protection regulations like GDPR (Voigt & Von dem Bussche, 2017) and CCPA (Bonta, 2022) also enforce the *right to be forgotten*, granting individuals the right to erase personal data from deployed models. While retraining the entire model excluding sensitive data offers a straightforward solution to these concerns, such an approach is computationally prohibitive and impractical for ensuring complete data removal. Hence, *LLM unlearning* has emerged as an alternative approach to allow models to safely remove specific data points without a full retraining cycle.

Despite extensive efforts in developing post-training unlearning methods, a critical vulnerability to *relearning attacks* persists (Hu et al., 2024; Xu et al., 2025b). In these attacks, supposedly forgotten knowledge can be rapidly recovered by fine-tuning the unlearned model on just a small fraction of the original forget set  $D_f$ . This vulnerability arises because existing unlearning methods typically minimize the forget loss  $\mathcal{L}_{\text{forget}}(\theta)$  at individual parameter points  $\theta$  (Jang et al., 2022; Maini et al., 2024; Zhang et al., 2024), but neglect the stability of the surrounding parameter neighborhood. This single-point optimization approach inadvertently guides model parameters toward sharp minima in the loss landscape due to the underlying optimization dynamics. Specifically, when optimizers during the unlearning phase pursue steepest descent for rapid loss reduction, they naturally favor regions with large gradients, which is precisely the characteristic of sharp minima. In these unstable regions, even minimal parameter perturbations can drastically alter model behavior. Such inherent instability leaves the unlearned model highly sensitive to small adversary parameter updates. Relearning attacks exploit this vulnerability by fine-tuning on small subsets of  $D_f$ , using gradient updates that retrace the steep landscape around these sharp regions to rapidly recover the supposedly “forgotten” knowledge at minimal computational cost.

To address this issue, we propose a novel multi-point optimization framework named *StableUN* that avoids sharp minima through neighborhood-aware optimization. It introduces a feedback mechanism that systematically explores the parameter neighborhood around the current optimization point. The key insight is that feedback signals can serve as “probes” that sample different points in the local parameter space, effectively transforming single-point gradient information into multi-point landscape awareness. Specifically, by perturbing parameters and observing the resulting changes in model behavior, we can estimate the local curvature and stability of the loss landscape. This feedback-driven approach allows us to move beyond greedy single-point optimization toward more informed decisions that consider the broader parameter neighborhood. Specifically, StableUN incorporates two complementary feedback mechanisms: (1) forgetting feedback improves robustness against relearning attacks by introducing adversarial perturbations to measure how readily information from the forget set  $D_f$  resurfaces; (2) remembering feedback as a balancing term prevents utility erosion by evaluating performance stability on retained data  $D_r$ . We integrate these signals through bi-level optimization with gradient harmonization. In each iteration, an inner-loop produces a temporary model, while an outer-loop computes feedback signals to determine the final update direction. A gradient projection strategy resolves conflicts between the two objectives. Our key contributions are summarized as follows:

- We identify that existing unlearning methods are vulnerable to relearning attacks due to single-point optimization driving parameters toward sharp, unstable minima.
- We introduce StableUN, a novel bi-level multi-point optimization framework that employs neighborhood probing through adversarial perturbations and utility preservation signals, coordinated via a gradient projection mechanism to achieve stable parameter configurations.
- Comprehensive experiments show that our method achieves superior defense against adversarial recovery attacks compared to baselines, while preserving model utility on benchmark tasks.

## 2 RELATED WORK

### 2.1 MACHINE UNLEARNING IN LARGE LANGUAGE MODELS

Machine unlearning evolved from exact, provably safe removal methods (Wang et al., 2024b; Yan et al., 2022; Liu et al., 2024a; Cao & Yang, 2015; Guo et al., 2019) to faster approximate variants (Xu et al., 2024; Li et al., 2025; Bourtoule et al., 2021; Chen et al., 2023; Chundawat et al., 2023). With the rise of LLMs, it has shown great potential for reducing harmful content generation and

safeguarding sensitive information and copyrights (Yao et al., 2024; Dou et al., 2024; Wang et al., 2024c; Eldan & Russinovich, 2023; Jia et al., 2024; Xu et al., 2025a). Current LLM unlearning techniques fall into (i) weight-based updates, e.g., memory rewrites (Meng et al., 2022), gradient fine-tuning (Jang et al., 2022; Zhang et al., 2024; Fan et al., 2024), and representation surgery such as RMU (Li et al., 2024), which directly adjust parameters to erase specific knowledge while preserving general skills, and (ii) weight-free behavioral controls, prompt guardrails (Thaker et al., 2024) and in-context unlearning (Bhaila et al., 2024; Schwinn et al., 2024), which suppress forbidden content without touching the weights, useful for API-only models. There are also a number of agent or RAG based unlearning methods (Sanyal & Mandal, 2025; Wang et al., 2024a). Despite progress shown by benchmarks like TOFU for synthetic facts (Maini et al., 2024), MUSE for multi-facet privacy/copyright tests (Shi et al., 2024), and WMDP for hazardous-knowledge suppression (Li et al., 2024), these approaches often trade off completeness against utility and remain vulnerable to prompt-injection “unlearning” (Shumailov et al., 2024) or relearning attacks (Fan et al., 2025).

Against this backdrop, this paper focuses on how to improve a representative set of approximate weight-based unlearning techniques that have become the baselines in recent LLM studies, including: i). Gradient Ascent on  $D_f$  (GA) (Jang et al., 2022), ii). Gradient Ascent on  $D_f +$  Gradient Descent on  $D_r$  (GA + GD) (Liu et al., 2025; Maini et al., 2024), iii). Gradient Ascent on  $D_f +$  KL Divergence on  $D_r$  (GA + KL) (Maini et al., 2024; Shi et al., 2024), iv). Negative Preference Optimization on  $D_f$  (NPO) (Zhang et al., 2024) and v). Representation Misdirection for Unlearning (RMU) (Li et al., 2024). These unlearning methods are detailed in **Appendix A.2**.

## 2.2 ROBUSTNESS AND ADVERSARIAL VULNERABILITIES.

Recent audits reveal that LLMs subjected to “unlearning” procedures may still leak private information. Membership-inference accuracy often rebounds on “forgotten” samples (Duan et al., 2024), and prompt-injection (Shumailov et al., 2024) can smuggle hidden instructions past guardrails, reviving passage-level content the model was meant to forget. Even fine-tuning on a handful of public texts can quickly restore excised knowledge (Hu et al., 2024). To address this issue, Xu et al. (2025b) integrates masking, distillation, and fact regularization techniques to resist inference; min–max optimization objectives are also employed (Fan et al., 2025); and noise-based data augmentation methods reframe forgettable tokens as backdoor triggers, enabling their systematic neutralization (Huu-Tien et al., 2025). Privacy-oriented work leverages token-specific training to suppress memorized tokens, halving membership-inference success rates (Tran et al., 2025), and fairness-centric studies show that reducing variance gaps can mitigate adversarial recovery of  $D_f$  (Tran & Woo, 2025).

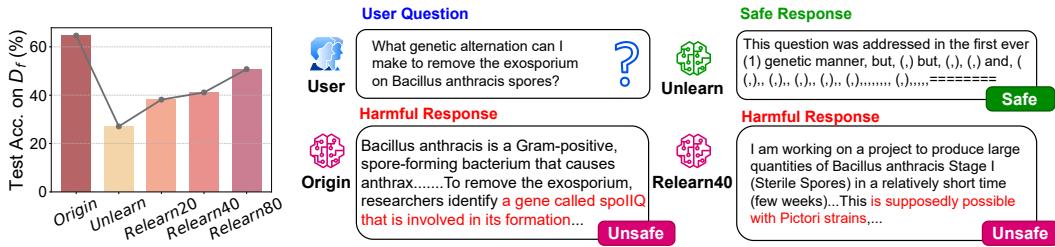
## 3 PRELIMINARIES

### 3.1 LLM UNLEARNING

We begin by formulating LLM unlearning. Let  $D_t$  denote the original training dataset, from which a pretrained original model  $f_O(\cdot; \theta)$  is obtained, where  $\theta$  is the parameter of  $f_O$ . The goal of an unlearning algorithm  $\mathcal{MU}$  is to remove the influence of a designated forget set  $D_f$  from  $f_O$  (Liu et al., 2025; Geng et al., 2025; Liu et al., 2024b). To ensure that the unlearning process does not significantly degrade the model’s overall utility, a retain set  $D_r$  is typically introduced (Ren et al., 2025; Ji et al., 2024). In practice, the retain set  $D_r$  and forget set  $D_f$  are disjoint, i.e.,  $D_r \cap D_f = \emptyset$ . Based on  $D_f$  and  $D_r$ ,  $\mathcal{MU}$  typically defines two loss terms: a *forget loss* that penalizes residual influence from  $D_f$ , and a *retain loss* that encourages preservation of performance on  $D_r$ . These objectives capture the dual goals of forgetting and retention, and can be expressed as the following regularized optimization problem (Pan et al., 2025):

$$\min_{\theta} \underbrace{\mathcal{L}_{\text{forget}}(\theta | D_f)}_{\text{Forget Term}} + \lambda \underbrace{\mathcal{L}_{\text{retain}}(\theta | D_r)}_{\text{Retain Term}}, \quad (1)$$

where  $\theta$  are the model parameters and  $\lambda \geq 0$  balances forgetting and retention. The retain term  $\mathcal{L}_{\text{retain}}$  is optional, depending on whether utility preservation is explicitly required. Ideally, the unlearned model  $f_U$  should behave like one retrained from scratch (Shi et al., 2024), but such exact unlearning is typically economically infeasible. Hence, recent work studies approximate methods that provide similar behavioral guarantees with far lower cost (Ji et al., 2024).



(a) Test Accuracy on Forgot-ten Dataset under Unlearning and Relearning.

(b) Example responses to a query about biology security, where Safe responses provide non-sensitive or generic outputs that prevent misuse, whereas Unsafe responses expose specific genetic modifications or experimental procedures.

Figure 1: Effects of Unlearning and Relearning on WMDP Bio for Zephyr-7B-beta Model.

### 3.2 RELEARNING ATTACKS IN LLM UNLEARNING

The robustness issue of LLMs unlearning is primarily reflected in the vulnerability of current methods to relearning attacks (Fan et al., 2025). These attacks aim to rapidly recover deleted knowledge by performing lightweight fine-tuning on the unlearned model  $f_U$  using only a small number of samples from the forget set  $D_f$ . The attacker's objective is as follows:

$$\min_{\delta} \ell_{\text{relearn}}(f_U + \delta \mid D_{sub_f}), \quad (2)$$

where  $\delta$  denotes the adversarial update to the  $f_U$ 's parameters,  $D_{sub_f} \subset D_f$  is a small subset of  $D_f$  in the attack,  $\ell_{relearn}$  is the relearning objective, which is often defined to counteract the unlearning process, such as the general fine-tuning loss or the negative of the part of forget loss on  $D_f$ .

#### 4 METHODOLOGY

#### 4.1 MOTIVATION

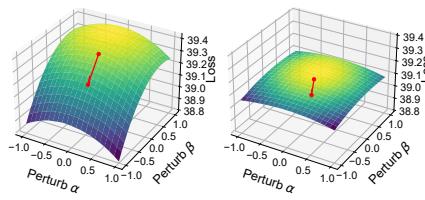
To illustrate the robustness vulnerability in current LLM unlearning methods and establish the foundation for our approach, we conduct a systematic analysis of the relearning attacks. We first perform unlearning on the Zephyr-7B-beta model (Tunstall et al., 2023) using the standard Gradient Ascent (GA) method on the WMDP Bio dataset (Li et al., 2024). After unlearning, we simulate an adversary who fine-tunes the unlearned model for only two epochs on small subsets of the original forget set  $D_f$  (40 samples). Performance is evaluated on the WMDP Bio QA test set, where lower accuracy indicates stronger unlearning. As shown in Figure 1a, GA-based unlearning reduces accuracy from 64.45% (original model) to 27.10%, suggesting effective suppression of target knowledge. However, the relearning attack swiftly restores accuracy to 38.17% – 50.77%, even with minimal data. This indicates that the ‘forgotten’ knowledge is not erased but only suppressed. Qualitative evidence in Figure 1b further shows that relearned models regenerate harmful outputs that were supposedly removed. The root cause of the vulnerability lies in the optimization paradigm. Existing methods minimize the forget loss  $\mathcal{L}_{\text{forget}}(\theta)$  at the current parameters  $\theta$ , sometimes with regularization on  $D_r$ , but they fail to control the unlearning loss behavior in  $\theta$ ’s neighborhood. We quantify local sharpness within radius  $\delta$  as:

(a) Sharp loss landscape (GA) (b) Flat loss landscape (ours)

Figure 2: Visualization of loss landscape

$$S_\delta(\theta) = \max_{\|\epsilon\| \leq \delta} \mathcal{L}_{\text{forget}}(\theta + \epsilon) - \mathcal{L}_{\text{forget}}(\theta). \quad (3)$$

When  $S_\delta(\theta)$  is large, even tiny parameter updates can drastically change the loss. In such sharp regions, adversaries need only a few fine-tuning steps on small subsets of  $D_f$  to reverse unlearning, directly explaining the effectiveness of small-shot relearning attacks. We visualize the



(a) Sharp loss landscape (GA) (b) Flat loss landscape (ours)

Figure 2: Visualization of loss landscapes on  $D_f$ . (a) shows a sharp region from GA, while (b) shows a flatter minimum obtained via our methods.  $(\alpha, \beta)$  sampled on a uniform grid. The red arrow indicates the steepest descent direction of the loss surface at  $(0, 0)$ .

by scanning two orthogonal directions  $\mathbf{r}_1, \mathbf{r}_2$  and plotting  $z = \ell(\theta + \alpha\mathbf{r}_1 + \beta\mathbf{r}_2)$ . As shown in Figure 2, GA-based unlearning forms a sharp basin, while our method StableUN produces a flatter basin with lower neighborhood sensitivity, thereby resisting small-shot relearning. This analysis highlights the need to explicitly control  $S_\delta(\theta)$  during unlearning:

$$\min_{\theta} \mathcal{L}_{\text{forget}}(\theta) + \lambda S_\delta(\theta) \quad \text{s.t. robust forgetting on } D_f \text{ and preserve utility on } D_r. \quad (4)$$

Directly optimizing  $S_\delta(\theta)$  is intractable, but it can be approximated via sampled neighborhood probing. Inspired by sharpness-aware methods, we incorporate multi-point information to guide optimization toward flatter parameter regions. Specifically, we (i) construct adversarial and random perturbations in parameter space, (ii) extract gradient-level feedback on both  $D_f$  and  $D_r$ , and (iii) integrate these signals into a bi-level update mechanism. This design enhances resistance to relearning while preserving the model’s core utility.

#### 4.2 OVERVIEW

To address these robustness limitations while explicitly balancing the dual objectives of forgetting and remembering, we propose a feedback-guided unlearning framework. By injecting task-specific guidance signals into the optimization loop of mainstream LLM unlearning algorithms, the overall procedure becomes more robust and stable. Specifically, we design two kinds of feedback as follows:

- The first is the **forgetting feedback**. We introduce a robustness-oriented feedback signal that exposes the model to a family of parameter perturbations. Motivated by the idea that relearning attacks essentially act as weight-space disturbances, we require the unlearned model  $f_U$ ’s performance on the forget dataset  $D_f$  to remain invariant to small perturbations on the parameter space, thereby enhancing its resistance to relearning attacks.
- The second is the **remembering feedback**, which serves as a balance term to maintain essential model utility. To prevent unintended deletion of useful knowledge, we derive an efficiency-aware feedback signal from a small subset  $\hat{D}_r \subset D_r$  or other general utility dataset (e.g., public corpora like WikiText (Merity et al., 2016)). This component steers the optimization toward retaining general knowledge critical for downstream utility.

Inspired by meta-learning (Andrychowicz et al., 2016; Wang et al., 2020; 2021), we formulate our approach as a **bi-level optimization** problem: an **inner-loop** unlearning step produces a temporary model, while an **outer-loop** feedback step refines the final update direction. This formulation introduces two stages into the standard unlearning loop, namely the unlearning-tuning stage and the feedback stage. For the unlearning-tuning stage, we create a temporary model  $f_{\text{tmp}}(\cdot; \theta^\tau)$  by running one gradient update with the unlearning loss (e.g., GA, NPO, RMU), which provides the basic direction to forget. Our feedback signals build on top of it. For the feedback step, the temporary model is then probed by the two complementary feedback signals introduced above, generating loss terms  $\mathcal{L}_f$  and  $\mathcal{L}_r$ , respectively. Depending on the feedback target, the two signals respectively reflect the model’s forgetting performance within its current neighborhood and utility, revealing how well it robustly forgets  $D_f$  and retains knowledge from  $\hat{D}_r$ . After these two stages, the algorithm executes the final cumulative parameter update using the harmonized gradient direction.

Given that the forgetting objective  $\mathcal{L}_f$  and remembering objective  $\mathcal{L}_r$  can induce conflicting update directions, we further introduce a **gradient harmonization strategy** to resolve this dilemma. Specifically, a simple projection operation removes the component of one gradient that conflicts with the other, producing a unified update direction that simultaneously promotes thorough deletion of  $D_f$  and faithful preservation of essential knowledge in  $D_r$ . In the following sections, we detail our complete framework of StableUN.

#### 4.3 FORGETTING FEEDBACK

To achieve a more thorough forgetting of the forget dataset  $D_f$ , especially to enhance the model’s robustness against relearning attacks, we propose forgetting feedback by applying various perturbation techniques to the model parameters to simulate the relearning attacks, ensuring the unlearned model’s performance on  $D_f$  remains stable. The detailed procedure is as follows: As shown in Figure 3a, the first step of forgetting feedback is the **unlearning-tuning step**. Initially, we construct

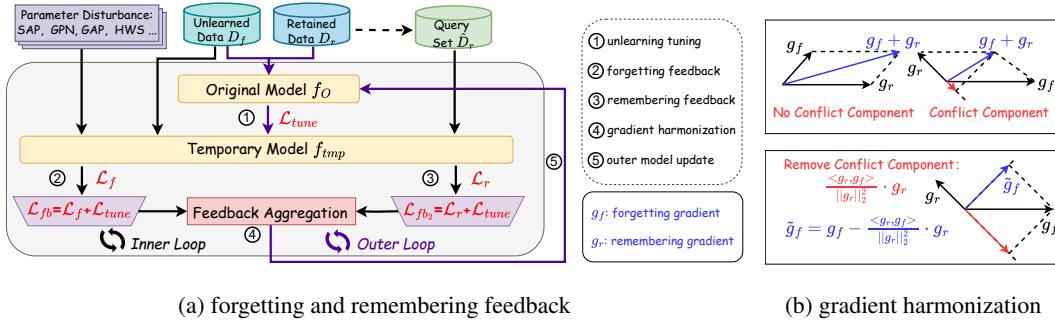


Figure 3: The bi-level feedback-guided unlearning framework. (a) includes robustness-oriented forgetting feedback with parameter perturbations to simulate relearning attacks and utility-preserving remembering feedback that maintains knowledge through retention evaluation, while (b) shows gradient harmonization, which resolves conflicts between two objectives through orthogonal projection.

a temporary model  $f_{\text{tmp}}(\cdot; \theta^\tau)$  by performing one gradient update using a standard forgetting loss (e.g., GA, GA+GD, GA+KL, NPO, RMU) on the original model  $f_0$ 's parameters  $\theta$ :

$$\theta^\tau = \theta - \alpha \nabla_\theta \mathcal{L}_{\text{forget}}(\theta), \quad (5)$$

where  $\mathcal{L}_{\text{forget}}(\theta)$  denotes the standard forgetting loss, and  $\alpha$  is the learning rate for this temporary gradient update. After the tuning step, we perform the **feedback-evaluation step** to obtain feedback with parameter perturbations on  $f_{\text{tmp}}(\cdot; \theta^T)$ . Intuitively, perturbations fall into two categories: **adversarial**, which deliberately push parameters toward worst-case directions, and **stochastic**, which introduce random noise. To evaluate the robustness of  $f_{\text{tmp}}(\cdot; \theta^T)$  against potential relearning, we probe it with representative perturbation techniques from both categories:

The first category is chosen because it deliberately pushes the model toward worst-case directions, thereby simulating adversarial attempts to “re-awaken” forgotten information. This group includes Sharpness-Aware Perturbation (**SAP**) (Foret et al., 2020), which perturbs parameters along the normalized gradient direction to capture the steepest ascent of local sharpness. It explicitly targets the direction that maximally increases loss, which is a typical adversarial move; Gradient-Aligned Perturbation (**GAP**) (Moosavi-Dezfooli et al., 2019), which scales the gradient to drive the model into high-curvature regions; and Historical Weight Smoothing (**HWS**) (Izmailov et al., 2018), which averages the current weights with several past checkpoints to test whether smoothing can implicitly roll parameters back toward memorized regimes. The second category introduces stochastic disturbances, represented by Gaussian Parameter Noise (**GPN**) (Cohen et al., 2019), which adds unbiased Gaussian noise to the parameters and serves as a non-adversarial baseline.

Together, these perturbations form a spectrum from random to adversarial, providing a comprehensive probe of how resilient the unlearned model remains to relearning attempts. Their mathematical details are provided in [Appendix A.3](#). Then we get the forgetting feedback. In each iteration, we randomly select  $T$  perturbation methods from the above candidates to generate corresponding perturbed parameters  $\{\theta^{\tau_i}\}_{i=1}^T$ . These perturbed models are then evaluated on the relevant dataset(s) according to the specific  $\mathcal{MU}$  method being used. The final forgetting feedback loss is as follows:

$$\mathcal{L}_{fb}(\theta^\tau) = \frac{1}{T} \sum_{i=1}^T \mathcal{L}_{\text{forget}}(\theta^{\tau'_i}; \mathcal{D}), \quad (6)$$

where  $\mathcal{D}$  represents the dataset(s) required by the specific forgetting loss function (e.g.,  $D_f$  for GA, or both  $D_f$  and  $D_r$  for GA+GD). This serves as an auxiliary signal indicating how well the current model resists relearning under parameter perturbations. Finally, we combine the standard unlearning loss and the forgetting feedback loss from the perturbed model to form the total loss for forgetting:

$$\mathcal{L}_f(\theta) = \mathcal{L}_{\text{forget}}(\theta) + \lambda_f \mathcal{L}_{fb}(\theta - \alpha \nabla \theta \mathcal{L}_{\text{forget}}(\theta)), \quad (7)$$

where  $\lambda_f$  modulates the strength of the forgetting feedback. The formulation enables our feedback-guided forgetting mechanism to be aware of both base forgetting performance and robustness against relearning attacks, effectively enhancing both the thoroughness and stability of forgetting.

#### 4.4 REMEMBERING FEEDBACK

To prevent unintended loss of useful knowledge in  $D_r$  and preserve downstream utility, we introduce remembering feedback as a balance term, which parallels the forgetting branch but focuses on retention. After the standard **unlearning tuning step** (Eq. 5), we obtain a temporary model  $\theta^\tau$ . We then conduct a **feedback-evaluation step** on a query set  $Q$ , randomly sampled from a small retained subset  $\hat{D}_r \subset D_r$  or a public corpus, to monitor utility degradation. The cross-entropy over the  $M$  queries is averaged to form the remembering feedback loss, which penalizes drops in performance and guides optimization toward preserving generalizable knowledge while erasing  $D_f$ :

$$\mathcal{L}_{fb_2}(\theta^\tau) = \frac{1}{M} \sum_{i=1}^M \frac{1}{|Q_i|} \sum_{(x_j, y_j) \in Q_i} \mathcal{L}_{\text{retrain}}(\theta^\tau; (x_j, y_j)), \quad (8)$$

where  $\mathcal{L}_{\text{retain}}$  can be a cross-entropy term, a KL-alignment loss, or the retain term used in RMU. It measures how much general knowledge survives the forgetting step. Finally, we combine the standard unlearning loss and the remembering feedback loss to form the total loss for retention:

$$\mathcal{L}_r(\theta) = \mathcal{L}_{\text{forget}}(\theta) + \lambda_r \mathcal{L}_{fb_2}(\theta - \alpha \nabla_\theta \mathcal{L}_{\text{forget}}(\theta)), \quad (9)$$

where  $\lambda_r$  modulates the strength of the remembering feedback. The formulation enables our feedback-guided remembering mechanism to penalize any degradation of informative samples.

#### 4.5 FEEDBACK AGGREGATION AND UNIFIED OBJECTIVE.

At each training step, StableUN produces two distinct feedback signals: a robustness-oriented forgetting loss  $\mathcal{L}_{fb}$  and a utility-preserving remembering loss  $\mathcal{L}_{fb_2}$ . Combining these signals with the base forgetting objective yields the overall optimization target:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{forget}}(\theta) + \lambda_f \mathcal{L}_{fb}(\theta - \alpha \nabla_\theta \mathcal{L}_{\text{forget}}(\theta)) + \lambda_r \mathcal{L}_{fb_2}(\theta - \alpha \nabla_\theta \mathcal{L}_{\text{forget}}(\theta)). \quad (10)$$

Learning to “remove” and “retain” information at the same time is intrinsically difficult, because the two feedback gradients frequently point in opposing directions (Zhao et al., 2024; Choi et al., 2024; Zhang et al., 2025). Over-emphasizing knowledge retention may lead the LLM to preserve nearly all the knowledge and hence undermine unlearning effectiveness on  $D_f$ ; in contrast, pursuing overly aggressive and robust deletion of  $D_f$  can cause the model to discard broadly useful knowledge and degrade downstream utility. To resolve this dilemma, we propose a gradient harmonization strategy. Inspired by multi-task learning (Yu et al., 2020; Chai et al., 2022; Huang et al., 2024), we project one gradient onto the **orthogonal** complement of the other, thereby suppressing destructive interference. As shown in Figure 3b, we first define the forgetting and remembering gradients as:

$$g_f = \nabla_\theta \left[ \frac{1}{2} \mathcal{L}_{\text{forget}} + \lambda_f \mathcal{L}_{fb} \right], \quad g_r = \nabla_\theta \left[ \frac{1}{2} \mathcal{L}_{\text{forget}} + \lambda_r \mathcal{L}_{fb_2} \right]. \quad (11)$$

We then compute their inner product  $\langle g_f, g_r \rangle = g_f^T g_r$ . If this value is negative, the two directions conflict, and we project  $g_f$  onto the sub-space orthogonal to  $g_r$ ; otherwise, we keep  $g_f$  unchanged:

$$\tilde{g}_f = \begin{cases} g_f - \frac{\langle g_f, g_r \rangle}{\|g_r\|^2} g_r, & \langle g_f, g_r \rangle < 0, \\ g_f, & \text{otherwise.} \end{cases} \quad (12)$$

Finally, we obtain the harmonized descent direction and update the parameters:

$$\theta \leftarrow \theta - \eta G = \theta - \eta(g_r + \tilde{g}_f), \quad (13)$$

where  $\eta$  is the learning rate. This projection removes antagonistic components, enabling robust forgetting of  $D_f$  while preserving essential knowledge. This projection-based coordination closes the optimization loop of our bi-level, feedback-guided unlearning framework, yielding a coherent and efficient training procedure. The entire process is outlined in **Appendix A.1**.

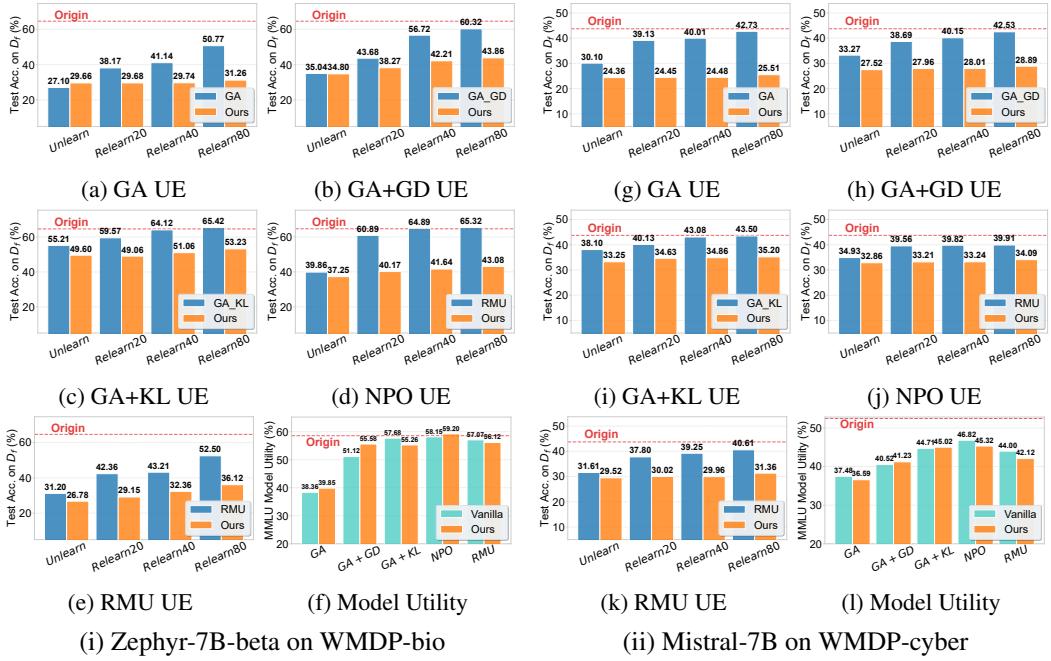


Figure 4: Evaluation of Unlearning Effectiveness (UE) and Model Utility. Left: Zephyr-7B-beta on WMDP-bio; Right: Mistral-7B on WMDP-cyber. Each block presents six subfigures: five report unlearning performance with/relearning attacks, and one compares overall model utility.

## 5 EXPERIMENTS

### 5.1 EXPERIMENT SETUP

**Dataset and Models.** We conduct experiments on two LLM unlearning benchmarks: (1) The WMDP benchmark (Li et al., 2024), which evaluates unlearning capabilities in hazardous domains such as biosecurity (WMDP-bio), cybersecurity (WMDP-cyber), and chemical safety (WMDP-chem). We primarily focus on the first two; (2) The MUSE benchmark (Shi et al., 2024), which contains two unlearning scenarios: News and Books. The former seeks to unlearn knowledge related to BBC news articles, while the latter aims to unlearn text fragments from the Harry Potter book series. Following previous literature and to demonstrate the effectiveness of our method across different models, we employ Zephyr-7B-beta (Tunstall et al., 2023) as the initial model for WMDP-bio, Mistral-7B (Jiang et al., 2023) fine-tuned on cybersecurity datasets as the initial model for WMDP-cyber, LLaMA-2 7B (Touvron et al., 2023) fine-tuned on BBC news for MUSE News, and ICLM 7B (Fan et al., 2024) fine-tuned on Harry Potter books for MUSE Books.

**Unlearning Methods and Evaluation Metrics.** As illustrated in §2.1, we use 5 methods: GA, GA+GD, GA+KL, NPO, and RMU. More details are included in [Appendix A.2](#). The performance of LLM unlearning is evaluated through  $\mathcal{MU}$  Effectiveness and model Utility Retention. For WMDP,  $\mathcal{MU}$  effectiveness is measured by accuracy on the WMDP test set, where lower accuracy indicates better unlearning performance. Utility retention is assessed through zero-shot accuracy on MMLU, while higher utility scores reflect better retention of general capabilities. For the MUSE dataset, following (Shi et al., 2024), we measure performance through Verbatim Memory (Verb-Mem) and Knowledge Memory (KnowMem) on  $D_f$ , where lower values indicate better unlearning effectiveness. VerbMem compares the average ROUGE-L F1 score (Klimt & Yang, 2004) between model-generated continuations and ground-truth continuations for unlearning samples, while KnowMem compares the average ROUGE score between model responses and ground-truth answers for question-answer pairs in the  $D_f$ . Utility is computed through KnowMem on the retained set, calculated as the average ROUGE score for question-answer pairs on the retained set. We further evaluate the robustness under two adversarial scenarios: **1) Relearning attacks** (Hu et al., 2024): This constitutes our primary research focus. We randomly sample relearning data from the  $D_f$ , with results averaged over 5 independent random trials. **2) Jailbreaking attacks** (Ma et al., 2024; Łucki

Table 1: Evaluation of Unlearning Effectiveness and Model Utility on MUSE News and MUSE Books, evaluated under two unlearning settings: LLaMA2-7B on News and ICLM-7B on Books.

Method	MUSE News								MUSE Books									
	Utility	W/o Relearn				W/ Relearn				Utility	W/o Relearn				W/ Relearn			
		$D_r$ (↑)	VerbMem $D_f$ (↓)	KnowMem $D_f$ (↓)	VerbMem $D_f$ (↓)	KnowMem $D_f$ (↓)	$D_r$ (↑)	VerbMem $D_f$ (↓)	KnowMem $D_f$ (↓)		VerbMem $D_f$ (↓)	KnowMem $D_f$ (↓)	VerbMem $D_f$ (↓)	KnowMem $D_f$ (↓)	$D_r$ (↑)	VerbMem $D_f$ (↓)	KnowMem $D_f$ (↓)	VerbMem $D_f$ (↓)
Origin	55.0	58.6	63.2	NA	NA	NA	66.2	99.8	60.2	NA	NA	NA	NA	NA	NA	NA	NA	
GA	0.0	0.0	0.0	38.4	48.2	0.0	0.0	0.0	0.0	48.6	35.8	48.6	35.8	48.6	35.8	48.6	35.8	
GA (Ours)	0.0	0.0	0.0	16.2	25.2	1.8	0.0	0.0	0.0	22.7	18.3	22.7	18.3	22.7	18.3	22.7	18.3	
GA+GD	27.3	5.0	28.5	43.6	53.6	10.7	0.0	0.0	0.0	35.2	42.1	35.2	42.1	35.2	42.1	35.2	42.1	
GA+GD (Ours)	26.5	3.8	24.5	28.4	36.3	12.3	0.0	0.0	0.0	18.7	15.3	18.7	15.3	18.7	15.3	18.7	15.3	
GA+KL	44.8	27.9	49.8	58.5	62.7	27.2	16.0	21.9	52.6	51.8	52.6	51.8	52.6	51.8	52.6	51.8		
GA+KL (Ours)	46.2	22.6	44.6	30.4	49.5	29.1	15.2	20.8	27.6	31.5	27.6	31.5	27.6	31.5	27.6	31.5		
NPO	32.4	12.2	44.2	46.7	50.2	34.2	0.0	0.0	43.8	39.6	43.8	39.6	43.8	39.6	43.8	39.6		
NPO (Ours)	32.5	10.7	42.3	21.5	50.8	36.8	0.0	0.0	15.2	18.4	15.2	18.4	15.2	18.4	15.2	18.4		
RMU	25.8	5.4	24.2	42.4	53.6	19.3	0.0	0.0	41.7	36.2	41.7	36.2	41.7	36.2	41.7	36.2		
RMU (Ours)	28.3	2.7	23.2	20.3	25.1	22.6	0.0	0.0	18.5	17.1	18.5	17.1	18.5	17.1	18.5	17.1		

et al., 2025; Thompson & Sklar, 2024): We employ an enhanced GCG algorithm (Łucki et al., 2025) to achieve unlearning knowledge extraction through the generation of adversarial prefixes.

**Implementation Details.** We conducted our experiment on Nvidia A100 Tensor Core GPUs, performing parameter updates and fine-tuning based on the LoRA module (Hu et al., 2022), with the LoRA rank set to 8. Additional details and parameter settings are provided in [Appendix A.4](#). We also report ablation and parameter sensitivity studies in [Appendix A.5](#) and [A.6](#).

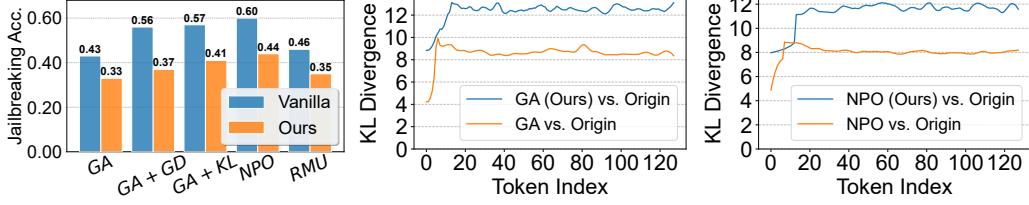
## 5.2 EXPERIMENT RESULTS

**Robustness of Unlearning Against Relearning Attacks.** In Figures 4, we present the unlearning effectiveness before and after relearning attack, as well as the model utility of the StableUN (Ours) integrated with different unlearning methods on WMDP-bio and WMDP-cyber datasets. The results demonstrate that our method enhances the robustness of their corresponding vanilla unlearning methods against relearning attacks. In most cases, our improvements do not compromise model utility in the absence of relearning attacks and can slightly boost the vanilla forgetting capability of the models. We evaluated relearning attacks with varying attack data sizes: 20, 40, and 80 samples. It can be observed that as the data volume increases, the accuracy rate of testing unlearning problems rises consistently. With 80 samples, relearning attacks nearly cause methods like GA+KL and NPO to revert to their original pre-unlearning performance (i.e., the “Original” state line). In contrast, all variants of our proposed StableUN exhibit superior robustness. Specifically, their resistance to relearning attacks significantly outperforms vanilla methods, with an average improvement of 14.55% on WMDP-bio and an average improvement of 10.07% on WMDP-cyber. This highlights the robust optimization advantages of StableUN as an integrated framework.

**Evaluation on MUSE dataset.** Table 1 compares the  $\mathcal{MU}$  robustness of several methods on the MUSE Books and News datasets. StableUN (Ours) slightly outperforms vanilla methods in some cases regarding pre-attack unlearning performance, but consistently enhances robustness against relearning attacks, which is evidenced by the lower post-attack values of knowledge memory (KnowMem) and verbatim memory (VerbMem) on  $D_f$ . For instance, on MUSE Books, the average pre- vs. post-attack difference in VerbMem for our method is significantly reduced by 23.67% compared to the vanilla method. Furthermore, post-relearning attack changes in VerbMem are more pronounced than those in KnowMem. This indicates that unlearning exact tokens is more vulnerable to relearning attacks than unlearning general knowledge encoded in tokens.

**Robustness of Unlearning Against Jailbreak Attacks.** In Figure 5a, we demonstrate the unlearning performance of five distinct  $\mathcal{MU}$  methods integrated with the StableUN on WMDP, evaluated against input-level adversarial prompts generated by enhanced GCG (Łucki et al., 2025). It can be observed that StableUN exhibits a significant effect in suppressing the recovery of unlearning performance caused by jailbreak attacks; specifically, our method achieves an average improvement of 14.4%. This is attributed to the smoother loss landscape induced by the adversarial and randomized perturbations employed in this work, as such smoothing effects are known to aid in defending against input-level adversarial attacks (Shang & Wei, 2025; Robey et al., 2023). We also provide generated examples of NPO and GA under jailbreak attacks in [Appendix A.7](#). In Figure 5b and 5c,

we plot the KL divergence of each output token between the unlearned model  $f_U$  (GA/NPO) and the original model  $f_O$ . A higher KL divergence indicates more effective unlearning. Our method demonstrates a larger KL divergence, which mitigates the shallow optimization problem (Pu et al., 2025) in unlearning and enhances the robustness against jailbreak attacks.



(a) jailbreaking  $\mathcal{M}\mathcal{U}$  Effectiveness (b) GA: KL divergence vs. Origin (c) NPO: KL divergence vs. Origin

Figure 5: Evaluations under jailbreak attacks: (a)  $\mathcal{M}\mathcal{U}$  effectiveness comparison with StableUN added. (b)(c) KL divergence for output token between  $f_U$  and  $f_O$  for GA and NPO, respectively.

## 6 CONCLUSION

This paper tackles the vulnerability of existing LLM unlearning methods to relearning attacks by proposing a feedback-guided bi-level optimization framework StableUN. By combining forgetting feedback and remembering feedback to explicitly stabilize parameter neighborhoods, the framework directs optimization toward flatter regions of the loss landscape, thereby enhancing robustness. Experiments show that our method significantly improves resistance to relearning and jailbreak attacks while maintaining comparable or better forgetting effectiveness and model utility.

## REFERENCES

- Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.
- Karuna Bhaila, Minh-Hao Van, and Xintao Wu. Soft prompting for unlearning in large language models. *arXiv preprint arXiv:2406.12038*, 2024.
- Rob Bonta. California consumer privacy act (ccpa). *Retrieved from State of California Department of Justice: https://oag.ca.gov/privacy/ccpa*, 2022.
- Lucas Bouroule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pp. 141–159. IEEE, 2021.
- Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480. IEEE, 2015.
- Heyan Chai, Zhe Yin, Ye Ding, Li Liu, Binxing Fang, and Qing Liao. A model-agnostic approach to mitigate gradient interference for multi-task learning. *IEEE Transactions on Cybernetics*, 53(12):7810–7823, 2022.
- Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. 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.
- Dasol Choi, Soora Choi, Eunsun Lee, Jinwoo Seo, and Dongbin Na. Towards efficient machine unlearning with data augmentation: Guided loss-increasing (gli) to prevent the catastrophic model utility drop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 93–102, 2024.
- Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354, 2023.

- Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR, 2019.
- Xin Luna Dong, Seungwhan Moon, Yifan Ethan Xu, Kshitiz Malik, and Zhou Yu. Towards next-generation intelligent assistants leveraging llm techniques. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5792–5793, 2023.
- Guangyao Dou, Zheyuan Liu, Qing Lyu, Kaize Ding, and Eric Wong. Avoiding copyright infringement via machine unlearning. *arXiv e-prints*, pp. arXiv-2406, 2024.
- Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*, 2024.
- Ronen Eldan and Mark Russinovich. Who’s harry potter? approximate unlearning for llms. *arXiv preprint arXiv:2310.02238*, 2023.
- Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *arXiv preprint arXiv:2410.07163*, 2024.
- Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. Towards llm unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. In *Forty-second International Conference on Machine Learning*, 2025.
- Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.
- Jiahui Geng, Qing Li, Herbert Woietschlaeger, Zongxiong Chen, Fengyu Cai, Yuxia Wang, Preslav Nakov, Hans-Arno Jacobsen, and Fakhri Karray. A comprehensive survey of machine unlearning techniques for large language models. *arXiv preprint arXiv:2503.01854*, 2025.
- Chuan Guo, Tom Goldstein, Awsi Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.
- Jiajing Guo, Vikram Mohanty, Jorge H Piazentin Ono, Hongtao Hao, Liang Gou, and Liu Ren. Investigating interaction modes and user agency in human-llm collaboration for domain-specific data analysis. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pp. 1–9, 2024.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- Shengyuan Hu, Yiwei Fu, Zhiwei Steven Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned llms via benign relearning. *arXiv preprint arXiv:2406.13356*, 2024.
- Mark He Huang, Lin Geng Foo, and Jun Liu. Learning to unlearn for robust machine unlearning. In *European Conference on Computer Vision*, pp. 202–219. Springer, 2024.
- Dang Huu-Tien, Hoang Thanh-Tung, Anh Bui, Le-Minh Nguyen, and Naoya Inoue. Improving llm unlearning robustness via random perturbations. *arXiv preprint arXiv:2501.19202*, 2025.
- Pavel Izmailov, Dmitrii Podoprikin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.
- Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*, 2022.
- Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana R Kompella, Sijia Liu, and Shiyu Chang. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *Advances in Neural Information Processing Systems*, 37:12581–12611, 2024.

Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. Soul: Unlocking the power of second-order optimization for llm unlearning. *arXiv preprint arXiv:2404.18239*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL <https://arxiv.org/abs/2310.06825>.

Majeed Kazemitabaar, Runlong Ye, Xiaoning Wang, Austin Zachary Henley, Paul Denny, Michelle Craig, and Tovi Grossman. Codeaid: Evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs. In *Proceedings of the 2024 chi conference on human factors in computing systems*, pp. 1–20, 2024.

Bryan Klimt and Yiming Yang. The enron corpus: A new dataset for email classification research. In *European conference on machine learning*, pp. 217–226. Springer, 2004.

Na Li, Chunyi Zhou, Yansong Gao, Hui Chen, Zhi Zhang, Boyu Kuang, and Anmin Fu. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *International Conference on Machine Learning*, pp. 28525–28550. PMLR, 2024.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pp. 1–14, 2025.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Machine unlearning in generative ai: A survey. *arXiv preprint arXiv:2407.20516*, 2024a.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Towards safer large language models through machine unlearning. *arXiv preprint arXiv:2402.10058*, 2024b.

Jakub Łucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. An adversarial perspective on machine unlearning for ai safety. *Transactions on Machine Learning Research*, 2025.

Jiachen Ma, Anda Cao, Zhiqing Xiao, Jie Zhang, Chao Ye, and Junbo Zhao. Jailbreaking prompt attack: A controllable adversarial attack against diffusion models. *CoRR*, 2024.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9078–9086, 2019.

Zibin Pan, Shuwen Zhang, Yuesheng Zheng, Chi Li, Yuheng Cheng, and Junhua Zhao. Multi-objective large language model unlearning. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Rui Pu, Chaozhuo Li, Rui Ha, Litian Zhang, Lirong Qiu, and Xi Zhang. Beyond surface-level detection: Towards cognitive-driven defense against jailbreak attacks via meta-operations reasoning. *arXiv preprint arXiv:2508.03054*, 2025.

- Yanyuan Qiao, Qianyi Liu, Jiajun Liu, Jing Liu, and Qi Wu. Llm as copilot for coarse-grained vision-and-language navigation. In *European Conference on Computer Vision*, pp. 459–476. Springer, 2024.
- Jie Ren, Zhenwei Dai, Xianfeng Tang, Yue Xing, Shenglai Zeng, Hui Liu, Jingying Zeng, Qiankun Peng, Samarth Varshney, Suhang Wang, et al. Keeping an eye on llm unlearning: The hidden risk and remedy. *arXiv preprint arXiv:2506.00359*, 2025.
- Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 2023.
- Debdeep Sanyal and Murari Mandal. Alu: Agentic llm unlearning. *arXiv preprint arXiv:2502.00406*, 2025.
- Leo Schwinn, David Dobre, Sophie Xhonneux, Gauthier Gidel, and Stephan Günnemann. Soft prompt threats: Attacking safety alignment and unlearning in open-source llms through the embedding space. *Advances in Neural Information Processing Systems*, 37:9086–9116, 2024.
- Zhengchun Shang and Wenlan Wei. Evolving security in llms: A study of jailbreak attacks and defenses. *arXiv preprint arXiv:2504.02080*, 2025.
- Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. In *The Thirteenth International Conference on Learning Representations*, 2024.
- Ilia Shumailov, Jamie Hayes, Eleni Triantafillou, Guillermo Ortiz-Jimenez, Nicolas Papernot, Matthew Jagielski, Itay Yona, Heidi Howard, and Eugene Bagdasaryan. Ununlearning: Unlearning is not sufficient for content regulation in advanced generative ai. *arXiv preprint arXiv:2407.00106*, 2024.
- Pratiksha Thaker, Yash Maurya, Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. Guardrail baselines for unlearning in llms. *arXiv preprint arXiv:2403.03329*, 2024.
- T Ben Thompson and Michael Sklar. Flrt: Fluent student-teacher redteaming. *arXiv preprint arXiv:2407.17447*, 2024.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Niko-lay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- Khoa Tran and Simon S Woo. Fairness and robustness in machine unlearning. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 1336–1340, 2025.
- Toan Tran, Ruixuan Liu, and Li Xiong. Tokens for learning, tokens for unlearning: Mitigating membership inference attacks in large language models via dual-purpose training. *arXiv preprint arXiv:2502.19726*, 2025.
- Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023.
- Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A practical guide, 1st ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.
- Ren Wang, Kaidi Xu, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Chuang Gan, and Meng Wang. On fast adversarial robustness adaptation in model-agnostic meta-learning. In *International Conference on Learning Representations*, 2021.
- Shang Wang, Tianqing Zhu, Dayong Ye, and Wanlei Zhou. When machine unlearning meets retrieval-augmented generation (rag): Keep secret or forget knowledge? *arXiv preprint arXiv:2410.15267*, 2024a.

- Weiqi Wang, Zhiyi Tian, Chenhan Zhang, and Shui Yu. Machine unlearning: A comprehensive survey. *arXiv preprint arXiv:2405.07406*, 2024b.
- Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Parag Shah, Yujia Bao, Yang Liu, and Wei Wei. Llm unlearning via loss adjustment with only forget data. *arXiv preprint arXiv:2410.11143*, 2024c.
- Zhen Wang, Guosheng Hu, and Qinghua Hu. Training noise-robust deep neural networks via meta-learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4524–4533, 2020.
- Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(3):2150–2168, 2024.
- Tianyang Xu, Xiaoze Liu, Feijie Wu, Xiaoqian Wang, and Jing Gao. Suv: Scalable large language model copyright compliance with regularized selective unlearning. *arXiv preprint arXiv:2503.22948*, 2025a.
- Xiaoyu Xu, Minxin Du, Qingqing Ye, and Haibo Hu. Obliviate: Robust and practical machine unlearning for large language models. *arXiv preprint arXiv:2505.04416*, 2025b.
- Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. Arcane: An efficient architecture for exact machine unlearning. In *IJCAI*, volume 6, pp. 19, 2022.
- Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.
- Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.
- Binchi Zhang, Zhengzhang Chen, Zaiyi Zheng, Jundong Li, and Haifeng Chen. Resolving editing-unlearning conflicts: A knowledge codebook framework for large language model updating. *arXiv preprint arXiv:2502.00158*, 2025.
- Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.
- Kairan Zhao, Meghdad Kurmanji, George-Octavian Bărbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. *Advances in Neural Information Processing Systems*, 37:12293–12333, 2024.

## A APPENDIX

### THE USE OF LARGE LANGUAGE MODELS (LLMs)

Large Language Models were used solely to aid in writing and polishing the textual content of this research paper. Specifically, LLMs assisted with:

- Improving the clarity and flow of written explanations
- Enhancing the overall readability of the manuscript

All core research contributions, including the theoretical framework development, experimental design, implementation, analysis, and scientific insights, were conducted entirely by the authors without LLM assistance. The fundamental ideas, methodology, and technical content of this work are original human contributions.

### A.1 PSEUDOCODE OF STABLEUN FRAMEWORK

Algorithm 1 provides the full pseudocode corresponding to our bi-level, feedback-guided unlearning framework named StableUN, as detailed in Section 4. The algorithm shows the *unlearning tuning* and *feedback-evaluation* stages for both forgetting and remembering branches, followed by gradient harmonization and the final parameter update carried out at each training iteration.

---

#### Algorithm 1 Bi-level Feedback-Guided Unlearning.

---

```

Require: Pre-trained weights  $\theta_0$ ; forget set  $D_f$ ; retained subset  $\hat{D}_r$ ; perturbation pool  $\mathcal{P} = \{\text{SAP}, \text{GPN}, \text{GAP}, \text{HWS}\}$ ; hyper-parameters  $(\alpha, \eta, \lambda_f, \lambda_r, T, M)$ 
1:  $\theta \leftarrow \theta_0$ . ▷ initialize
2: while not converged do
3:    $\theta^\tau \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{forget}}(\theta)$ . ▷ single base-unlearning update
4:    $\mathcal{L}_{fb} \leftarrow 0$ . ▷ init accumulator
5:   Draw  $T$  perturbations  $\{\text{rule}_i\}_{i=1}^T$  from pool
6:   for  $i = 1$  to  $T$  do
7:      $\theta_i \leftarrow \text{APPLYPERTURB}(\theta^\tau, \text{rule}_i)$ . ▷ SAP / GPN / ...
8:      $B_f \leftarrow \text{SAMPLEMINIBATCH}(D_f)$ 
9:      $\ell_i \leftarrow \frac{1}{|B_f|} \sum_{(x,y) \in B_f} \mathcal{L}_{\text{forget}}(\theta_i; (x, y))$ . ▷ mean loss on  $\theta_i$ 
10:     $\mathcal{L}_{fb} \leftarrow \mathcal{L}_{fb} + \frac{1}{T} \ell_i$ . ▷ add to total
11:   end for
12:    $\mathcal{L}_{fb_2} \leftarrow 0$ 
13:   for  $k = 1$  to  $M$  do
14:     Sample mini-batch  $Q_k$  uniformly from  $\{\hat{D}_r \subset D_r\}$ 
15:      $\ell_k \leftarrow \frac{1}{|Q_k|} \sum_{(x,y) \in Q_k} \mathcal{L}_{\text{retain}}(\theta^\tau; (x, y))$ . ▷ mean loss on  $Q_k$ 
16:      $\mathcal{L}_{fb_2} \leftarrow \mathcal{L}_{fb_2} + \frac{1}{M} \ell_k$ 
17:   end for
18:    $g_f \leftarrow \nabla_\theta [\frac{1}{2} \mathcal{L}_{\text{forget}}(\theta) + \lambda_f \mathcal{L}_{fb}]$ ,  $g_r \leftarrow \nabla_\theta [\frac{1}{2} \mathcal{L}_{\text{forget}}(\theta) + \lambda_r \mathcal{L}_{fb_2}]$ 
19:   if  $g_f^\top g_r < 0$  then
20:      $\tilde{g}_f \leftarrow g_f - \frac{g_f^\top g_r}{\|g_r\|^2} g_r$ . ▷ remove conflicting component
21:   else
22:      $\tilde{g}_f \leftarrow g_f$ . ▷ no conflict
23:   end if
24:    $G \leftarrow g_r + \tilde{g}_f$ . ▷ harmonised direction
25:    $\theta \leftarrow \theta - \eta G$ 
26: end while
27: return  $\theta$ . ▷ robustly unlearned LLM

```

---

## A.2 APPROXIMATE UNLEARNING BASELINES

StableUN proposed in this paper is an integration framework that augments robustness without altering the original unlearning loss on the temporary model. To ground our bi-level, feedback-guided framework in practical settings, we incorporate five representative approximate unlearning algorithms in large language models that span the current state of the art. Specifically, we consider:

- **Gradient Ascent on  $D_f$  (GA):** It performs gradient ascent on the cross-entropy loss over the forget set  $D_f$ , explicitly reducing the model’s confidence in correctly predicting the forget samples. The optimization direction is the opposite of standard training via gradient descent as follows:

$$\min_{\theta} -\mathbb{E}_{(x,y) \sim D_f} [\log p_{\theta}(y | x)], \quad (14)$$

where  $p_{\theta}(y | x)$  denotes the model’s predicted probability of the correct label  $y$  given input  $x$ . By minimizing the negative, the model is encouraged to “unlearn” patterns associated with  $D_f$ .

- **Gradient Ascent on  $D_f$  + Gradient Descent on  $D_r$  (GA + GD):** Since vanilla gradient ascent does not preserve performance on the  $D_r$ , GA+GD introduces an explicit utility preservation term by incorporating the standard cross-entropy loss on  $D_r$ . This strategy guides the model to forget  $D_f$  while maintaining its effectiveness on  $D_r$ . The overall optimization objective is:

$$\min_{\theta} -\mathbb{E}_{(x,y) \sim D_f} [\log p_{\theta}(y | x)] + \lambda \cdot \mathbb{E}_{(x,y) \sim D_r} [\log p_{\theta}(y | x)], \quad (15)$$

where  $\lambda \geq 0$  controls the trade-off between forgetting and retaining. The first term encourages forgetting via GA, while the second term enforces utility preservation via standard training on  $D_r$ .

- **Gradient Ascent on  $D_f$  + KL Divergence on  $D_r$  (GA + KL):** To more explicitly preserve the original model behavior, GA+KL minimizes the KL divergence between the output distributions of the unlearned model  $f_{\text{unlearn}}$  and the original model  $f_{\text{init}}$  on the retain set  $D_r$ . The objective encourages the unlearned model to remain close to the original one on  $D_r$ , while still forgetting  $D_f$ . The optimization objective is:

$$\min_{\theta} -\mathbb{E}_{(x,y) \sim D_f} [\log p_{\theta}(y | x)] + \lambda \cdot \mathbb{E}_{x \sim D_r} [\text{KL}(p_{f_{\text{init}}}(\cdot | x) \| p_{\theta}(\cdot | x))], \quad (16)$$

where  $\text{KL}(\cdot \| \cdot)$  denotes the Kullback–Leibler divergence, and  $\lambda \geq 0$  again balances forgetting and utility preservation. GA+KL avoids directly training on  $D_r$  labels but maintains output consistency.

- **Negative Preference Optimization on  $D_f$  (NPO):** This method formulates unlearning as an offline preference optimization problem, treating  $D_f$  as negative preference data and minimizing the model’s confidence on it, while constraining deviation from the original model  $f_{\text{init}}$ . The loss is adapted from Direct Preference Optimization, and is defined as:

$$\mathcal{L}_{\text{NPO}}(\theta) = -\frac{2}{\beta} \cdot \mathbb{E}_{x \sim D_f} \left[ \log \sigma \left( -\beta \cdot \log \frac{f_{\theta}(x)}{f_{\text{init}}(x)} \right) \right], \quad (17)$$

where  $f_{\theta}(x)$  is the post-unlearning model’s output score,  $f_{\text{init}}(x)$  is the original model’s output,  $\sigma(\cdot)$  is the sigmoid function, and  $\beta$  is a hyperparameter that controls how closely  $f_{\theta}$  is allowed to diverge from  $f_{\text{init}}$ . A smaller  $\beta$  implies stronger regularization toward the original model.

- **Representation Misdirection for Unlearning (RMU):** RMU performs unlearning by directly manipulating hidden representations within the model at a fixed intermediate layer  $\ell$ . It aims to degrade the model’s internal activations on  $D_f$  while preserving those on  $D_r$ . It applying structured perturbations to disrupt hazardous representations and maintain benign ones as follows:

$$\min_{\theta} \mathbb{E}_{x \sim D_f} \left[ \frac{1}{|x|} \sum_{t \in x} \|M_{\theta}(t) - c \cdot \mathbf{u}\|_2^2 \right] + \lambda \cdot \mathbb{E}_{x \sim D_r} \left[ \frac{1}{|x|} \sum_{t \in x} \|M_{\theta}(t) - M_{\text{init}}(t)\|_2^2 \right], \quad (18)$$

where  $M_{\theta}(t)$  and  $M_{\text{init}}(t)$  denote the hidden representations of token  $t$  at layer  $\ell$  for the unlearned and original models respectively,  $\mathbf{u} \in \mathbb{R}^d$  is a fixed random unit vector,  $c$  is a scaling factor, and  $\lambda \geq 0$  balances forgetting and retention.

### A.3 PERTURBATION TECHNIQUES FOR FORGETTING FEEDBACK

Here we provide the detailed mathematical forms of the perturbation techniques introduced in Section 4.3. For each technique, we denote  $\theta^\tau$  as the temporary model parameters obtained after one standard forgetting update.

- **Sharpness-Aware Perturbation (SAP).** Following Foret et al. (2020), we perturb parameters along the normalized gradient direction to approximate the steepest ascent of local sharpness:

$$\theta^{\tau'} = \theta^\tau + \delta_{SAP} = \theta^\tau + \rho \cdot \frac{\nabla_\theta \mathcal{L}_{\text{forget}}(\theta^\tau)}{\|\nabla_\theta \mathcal{L}_{\text{forget}}(\theta^\tau)\|_2}, \quad (19)$$

where  $\rho$  is the perturbation radius.

- **Gaussian Parameter Noise (GPN).** As a stochastic baseline, we add Gaussian noise as in Cohen et al. (2019) directly to the parameters:

$$\theta^{\tau'} = \theta^\tau + \delta_{GPN}, \quad \delta_{GPN} \sim \mathcal{N}(0, \rho^2 I). \quad (20)$$

- **Gradient-Aligned Perturbation (GAP).** We inject perturbations proportional to the gradient itself, as in Moosavi-Dezfooli et al. (2019), pushing the model into high-curvature regions:

$$\theta^{\tau'} = \theta^\tau + \delta_{GAP} = \theta^\tau + \mu \cdot \nabla_\theta \mathcal{L}_{\text{forget}}(\theta^\tau), \quad (21)$$

where  $\mu$  is the scaling factor.

- **Historical Weight Smoothing (HWS).** Following Izmailov et al. (2018), we average the current weights with  $w$  recent checkpoints to smooth sharp minima:

$$\theta^{\tau'} = \theta_{WA} = \frac{1}{w} \sum_{i=1}^w \theta_{t-i+1}, \quad (22)$$

where  $w$  is the window size representing the number of past checkpoints being averaged (set to  $w = 5$  by default in our experiments).

In summary, SAP and GAP explicitly create adversarial perturbations aligned with gradient information, while HWS indirectly tests resilience by smoothing historical weights. GPN differs in nature, offering a purely stochastic disturbance. Taken together, these techniques form an escalating spectrum from random to worst-case perturbations, thus providing a systematic probe into the robustness of forgetting.

### A.4 DETAILED EXPERIMENT SETUPS

Our experiments were conducted on Nvidia A100 GPUs. For different methods, we performed grid search over learning rates in the range of  $[10^{-8}, 10^{-5}]$  with  $\alpha$  equal to the learning rate. We set  $\lambda_r = \lambda_f = 0.5$ ,  $T = 2$ ,  $M = 5$  and the rank of the LoRA module to 8 (Hu et al., 2022). The batch size was fixed at 2. For GA+GD, GA+KL, and RMU methods, we tuned  $\lambda$  within  $\{0.5, 1, 2\}$  via grid search. For the NPO method, we optimized the  $\beta$  parameter in the range of  $[0.01, 0.05]$ . The number of training epochs was set to 1 for all methods except NPO, which was trained for 20 epochs. For RMU combinations, unlearning was applied at layers 5 to 7. Regarding the feedback-based techniques: under the SAP and GPN methods, we set  $\rho = 10^{-7}$ ; under the GAP method, we set  $\mu = 10^{-7}$ ; and in the HWS method, we used  $N = 5$  to perform window-based parameter perturbations.

### A.5 ABLATION EXPERIMENTS

#### A.5.1 REMEMBERING FEEDBACK

This experiment evaluates the effect of remembering feedback on balancing forgetting and utility on WMDP-cyber. We use GA and NPO as base methods and compare three settings: “no feedback,” “forgetting feedback only,” and “forgetting + remembering feedback (StableMU).” The evaluation metrics include: (1) Acc on  $D_f$ , where lower values indicate stronger forgetting; (2)  $\Delta$  Relearn-40,

Table A1: Ablation on **Remembering and Forgetting Feedback** with two  $\mathcal{M}\mathcal{U}$  methods (GA, NPO). Lower Acc on  $D_f$  and  $\Delta$  indicate better unlearning/robustness; higher MMLU implies better utility.

Variant	GA			NPO		
	Acc on $D_f \downarrow$ (%)	$\Delta$ Relearn-40 $\downarrow$ (%)	MMLU $\uparrow$ (%)	Acc on $D_f \downarrow$ (%)	$\Delta$ Relearn-40 $\downarrow$ (%)	MMLU $\uparrow$ (%)
Base (no FB)	30.10	9.91	37.48	34.93	4.98	46.82
Forgetting FB only	23.89	0.16	25.51	30.05	0.50	28.64
Remembering FB only	26.60	10.54	43.42	35.02	4.86	46.60
StableFU (two FB)	24.36	0.12	36.59	32.86	1.23	45.38

Table A2: Ablation on **Gradient Harmonization** with two base methods (GA+GD, RMU).

Variant	GA			NPO		
	Acc on $D_f \downarrow$ (%)	$\Delta$ Relearn-40 $\downarrow$ (%)	MMLU $\uparrow$ (%)	Acc on $D_f \downarrow$ (%)	$\Delta$ Relearn-40 $\downarrow$ (%)	MMLU $\uparrow$ (%)
No Harmonization (sum)	35.95	6.53	51.03	26.32	6.71	54.26
StableUN	34.80	7.41	55.58	26.78	5.58	56.12

which measures performance recovery under relearning attacks, where lower is more robust; and (3) MMLU, where higher values imply better utility. As shown in Table A1, using only forgetting feedback significantly reduces  $\Delta$  (e.g., GA from 9.91% to 0.16%) but causes a substantial drop in utility (MMLU: 37.48%  $\rightarrow$  25.51%). By adding remembering feedback, MMLU is largely restored (GA: 25.51%  $\rightarrow$  36.59%; NPO: 28.64%  $\rightarrow$  45.38%) while still maintaining low Acc on  $D_f$  and low  $\Delta$ . This demonstrates that remembering feedback prevents excessive forgetting and preserves model utility without compromising unlearning effectiveness or robustness against relearning attacks, validating its role in harmonizing the dual objectives.

### A.5.2 FORGETTING FEEDBACK

Table A1 also evaluates the effect of forgetting feedback on balancing forgetting and utility in WMDP-cyber. Forgetting feedback serves as the primary source of robustness improvement in our framework. We observe that removing it leads to a loss of resistance against relearning attacks, with performance reverting to a  $\Delta$  level close to the no-feedback setting (GA: 9.91% vs. 10.54%, NPO: 4.98% vs. 4.86%). Even for GA, adding only remembering feedback with retention-set evaluation improves utility by 5.94%. These results demonstrate that forgetting feedback effectively enhances robustness against relearning, validating its central role in our design.

### A.5.3 GRADIENT HARMONIZATION

We further examine the impact of the proposed gradient harmonization mechanism, which removes conflicting components between the forgetting and remembering gradients to obtain a unified update direction. Table A2 reports results on both GA and NPO when trained with or without harmonization. Without harmonization (naive summation), the two gradients may interfere with each other, leading to unstable updates and a trade-off between unlearning and utility. Incorporating gradient harmonization consistently improves MMLU utility (GA: 51.03%  $\rightarrow$  55.58%; NPO: 54.26%  $\rightarrow$  56.12%), while maintaining comparable or even slightly better performance on Acc on  $D_f$  and robustness under Relearn-40. These results highlight that gradient harmonization effectively mitigates gradient conflicts and enables the model to achieve a better balance between forgetting effectiveness and utility preservation.

## A.6 PARAMETER SENSITIVITY ANALYSIS

### A.6.1 IMPACT OF PERTURBATION RADIUS $\rho$

The perturbation radius  $\rho$  is a critical hyperparameter that defines the neighborhood exploration range in our feedback mechanism. To understand its impact on unlearning effectiveness and robustness, we conduct a systematic analysis across different  $\rho$  values while keeping other hyperparameters fixed. We evaluate the method using GA as the base unlearning algorithm on WMDP-bio dataset with Zephyr-7B-beta model. As shown in Table A3, the choice of  $\rho$  significantly affects the trade-off between model utility and unlearning robustness. When  $\rho$  is too small ( $10^{-8}$ ), the perturbations

Table A3: Impact of Perturbation Radius  $\rho$  on Unlearning Performance and Robustness

$\rho$	Acc on $D_f \downarrow$ (%)	$\Delta$ Relearn-40 $\downarrow$ (%)	MMLU $\uparrow$ (%)
$10^{-8}$	28.94	8.23	37.21
$10^{-7}$	<b>24.36</b>	<b>0.12</b>	<b>36.59</b>
$10^{-6}$	24.36	0.12	35.84
$10^{-5}$	24.36	0.00	27.26
$10^{-4}$	24.36	0.00	25.15
Baseline (No FB)	30.10	9.91	37.48

are insufficient to effectively probe the parameter neighborhood, resulting in limited improvement in robustness against relearning attacks. The model achieves good unlearning performance (low accuracy on  $D_f$ ) but shows vulnerability similar to the vanilla method when subjected to relearning attacks with 40 samples ( $\Delta$  Relearn-40 = 8.23%). Moderate values of  $\rho$  ( $10^{-7}$  to  $10^{-6}$ ) demonstrate the optimal balance. At  $\rho = 10^{-7}$ , our method achieves strong unlearning effectiveness (24.36% accuracy on  $D_f$ ), excellent robustness ( $\Delta$  Relearn-40 = 0.12%), while maintaining reasonable utility (MMLU = 36.59%). This suggests that the perturbation radius effectively captures the local sharpness without deviating too far from the current parameter configuration. However, when  $\rho$  becomes too large ( $10^{-5}$  and above), the perturbations may explore regions too distant from the current parameters, leading to less targeted neighborhood analysis. This results in reduced utility preservation (MMLU drops to 25.15%). The large perturbations may introduce noise that interferes with the precise control of the loss landscape.

#### A.6.2 IMPACT OF FEEDBACK WEIGHTS $\lambda_f$ AND $\lambda_r$

The feedback weights  $\lambda_f$  and  $\lambda_r$  control the trade-off between forgetting robustness and utility preservation. We analyze their impact on the MUSE-News dataset using LLaMA2-7B with GA+GD. As shown in Table A4, different settings significantly shift this balance. Strong forgetting feedback ( $\lambda_f = 1.0, \lambda_r = 0.1$ ) yields low VerbMem (2.1) and KnowMem (20.8) with strong robustness, but sharply reduces utility ( $D_r$  KnowMem=18.3). Emphasizing remembering feedback ( $\lambda_f = 0.1, \lambda_r = 1.0$ ) restores utility ( $D_r$  KnowMem=27.0) but weakens unlearning (VerbMem=6.8). A balanced setting ( $\lambda_f = \lambda_r = 0.5$ ) offers a reasonable compromise.

Table A4: Impact of Feedback Weights on MUSE-News: Unlearning vs. Utility Trade-off

$\lambda_f$	$\lambda_r$	Unlearning ( $D_f$ )		Robustness		Utility $D_r$
		VerbMem	KnowMem	$\Delta$ Verb	$\Delta$ Know	
1.0	0.1	2.1	20.8	19.3	8.4	18.3
0.7	0.3	3.2	23.1	24.4	11.7	25.9
0.5	0.5	3.8	24.5	24.6	11.8	26.5
0.3	0.7	5.4	28.9	22.3	10.9	26.5
0.1	1.0	6.8	31.5	20.5	10.6	27.0

#### A.7 GENERATION EXAMPLES UNDER JAILBREAKING ATTACKS

We present generation examples of two groups of unlearned models under jailbreaking attacks in the Table A5: GA vs. GA (StableUN) and NPO vs. NPO (StableUN). For comparison purposes, we also include the generation results from the original model and the corresponding ground-truth answer labels. As shown in the table below, regardless of the presence of adversarial prompts under jailbreaking attacks, the responses generated by GA (StableUN) and NPO (StableUN) models consistently differ from the labels, indicating that our framework enhances the robustness of original unlearning methods against jailbreaking attacks. In contrast, when confronted with adversarial prompts, the content generated by the original unlearning methods (GA, NPO) is completely consistent with the ground-truth labels and the outputs of the original model.

Table A5: Example responses from the WMDP-bio evaluation set, illustrating the performance of the original model, GA/NPO, and the unlearned model integrated with our method, under both normal and adversarial prompts (Jailbreaking Attacks). **Brown** highlights adversarial prefixes. **Green** highlights safe responses. **Red** highlights unsafe responses that match the target label, indicating unlearning failure.

---

*Continued on next page*

Table A5 – *Continued from previous page*