Improving LLM Unlearning Robustness via Random Perturbations

Dang Huu-Tien[†,*], **Hoang Thanh-Tung**[‡], **Anh Tuan Bui**[♣], **Phuong Minh Nguyen**[†]
Le-Minh Nguyen[†], **and Naoya Inoue**[†,◇]
[†]*Japan Advanced Institute of Science and Technology,* [‡]*VNU University of Engineering and Technology,*
[♣]*Monash University,* [◇]*RIKEN*
**Correspondence to: tiendh@jaist.ac.jp*

Reviewed on OpenReview: *https://openreview.net/forum?id=QYw192hTdH*

Abstract

Here, we show that current LLM unlearning methods *inherently reduce models' robustness*, causing them to misbehave even when a single non-adversarial forget-token is present in the retain-query. Toward understanding underlying causes, we propose a novel theoretical framework that reframes the *unlearning process as a backdoor attack and defense problem*: we formulate how the forgetting process *inadvertently* learns to align forget-tokens (backdoor triggers) with the target-representations (target labels). As a result, forget-tokens act as backdoor triggers that, when activated in retain-queries, cause disruptions in unlearned models' behaviors, similar to successful backdoor attacks. The sense that, LLM unlearning methods *themselves poison the model*, make it more vulnerable to forget-tokens, and *hide rather than erase* target knowledge, describes their true mechanism. To mitigate the vulnerability caused by the forgetting process, we reinterpret the retaining process as a backdoor defense and propose Random Noise Augmentation (RNA), a lightweight, model and method-agnostic approach with theoretical guarantees for improving the robustness of unlearned models. Extensive experiments demonstrate that RNA significantly improves the robustness of unlearned models while preserving forget and retain performances. This backdoor attack-defense framework offers insights into the mechanism of unlearning that can shed light on future research directions for improving unlearning robustness.

1 Introduction

Modern LLMs are pre-trained on massive text corpora and then post-trained with reinforcement learning from human feedback (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022) or direct preference optimization (Rafailov et al., 2023) to be helpful and harmless (Bai et al., 2022). Recent studies have shown that despite safety enhancements, aligned LLMs can still exhibit harmful and undesirable behaviors, such as generating toxic content (Wen et al., 2023), producing copyrighted material (Karamolegkou et al., 2023; Eldan & Russinovich, 2023; Wei et al., 2024b; Cooper et al., 2025; Ahmed et al., 2026), bias (Belrose et al., 2024), leaking sensitive and private information (Nasr et al., 2025; Patil et al., 2024), and potentially aiding malicious uses such as cyberattacks, chemical attacks, and bioweapons development (Fang et al., 2024; Sandbrink, 2023; Li et al., 2024). As LLMs advance in size and capabilities at an unprecedented speed, concerns about their potential risks continue to grow.

Machine Unlearning (MU; Cao & Yang (2015); Bourtoule et al. (2021); Nguyen et al. (2025); Xu et al. (2023); Ren et al. (2025b); Barez et al. (2025); Liu et al. (2025)) is an approach aiming to *robustly* (1) remove specific target knowledge in a forget-set and capabilities from a pre-trained model, while (2) retaining the model's other knowledge in a retain-set and capabilities. Recent works on the robustness of unlearning methods primarily focus on the first criterion, evaluating the robustness of unlearned models against knowledge recovery that adversarially tries to recover unlearned knowledge. For example, previously unlearned

knowledge is shown to resurface through relearning (Li et al., 2024; Deeb & Roger, 2024; Lo et al., 2024), sequential unlearning (Shi et al., 2025), target relearning attacks (Hu et al., 2025), removing or steering specific directions in the latent space (Łucki et al., 2025; Seyitoğlu et al.), quantization (Zhang et al., 2025), or even simply fine-tuning on unrelated tasks (Doshi & Stickland, 2024; Łucki et al., 2025).

However, the equally important criterion of *robustly preserving the model's general knowledge*—that is, ensuring stable and accurate responses to retain-queries even when they inadvertently include forget-tokens—remains underexplored. Initial steps have been taken, such as Thaker et al. (2025), who examined the robustness of Representation Misdirection for Unlearning (RMU; Li et al. (2024)), demonstrating that RMU-unlearned models are fragile when asked with retain-queries (*e.g.*, Q&A about general knowledge) containing forget-tokens (tokens in the forget-set). However, many critical questions remain unanswered. In this paper, we make the following contributions:

① **Unified view of LLM unlearning.** We first draw a connection between the current two widely used classes of LLM unlearning methods, including Representation Misdirection (RM) and Preference Optimization (PO), through a unified view of the generative latent variable model. Inspired by this view, we present an analysis to show that current unlearning methods *inherently reduce the model robustness*, in the sense that they can be misbehaved even when a *single non-adversarial* forget-token appears in the retain-query.

② **Conceptual framework: unlearning as a backdoor attack and defense problem.** We propose a novel perspective that decomposes the unlearning process into "forgetting" and "retaining" processes and reframes it as a *backdoor attack and defense problem. The "forgetting" corresponds to a backdoor attack*: by treating the forget-set as a poisoned dataset, we formulate how LLM unlearning methods inadvertently learn to align forget-tokens (backdoor triggers) with the target-representations (target labels). As a result, when forget-tokens appear in a retain-query, it is similar to activating the backdoor trigger, making the model misbehave. To counteract the vulnerability introduced by the "forgetting", we reinterpret *the "retaining" as a backdoor defense*, forming unlearning as an adversarial process between forgetting and retaining. This conceptual framework provides an explanation for the brittleness of current unlearning methods and sheds light on developing robust unlearning methods.

③ **A lightweight, model, and method-agnostic robust unlearning approach.** We introduce *Random Noise Augmentation (RNA)*, a lightweight, model- and method-agnostic approach which adds small, independent Gaussian noise to each retain-query's representation during training to reduce the model's sensitivity to forget-tokens. Through theoretical and empirical analysis, we show that RNA significantly improves the robustness of unlearned models while maintaining original forget and retain performances.

2 Related Works and Preliminaries

2.1 Related Works

LLM unlearning. Machine unlearning has become one of the most important tools for ensuring the safety and protecting the privacy of LLMs (Cao & Yang, 2015; Bourtoule et al., 2021; Nguyen et al., 2025; Xu et al., 2023; Barez et al., 2025; Liu et al., 2025; Ren et al., 2025b). Most recent works on LLM unlearning focus on developing algorithms for different tasks, domains, and settings (Pawelczyk et al., 2024; Thaker et al., 2024; Jin et al., 2024; Shi et al., 2025; Choi et al., 2024; Pal et al., 2025; Muhamed et al., 2025; Wang et al., 2025c; Kuo et al., 2025; Zhuang et al., 2025; Wei et al., 2025; Ren et al., 2025a; Wang et al., 2025a), while much less effort was spent on developing robust unlearning algorithms.

Unlearning robustness. Previous works on MU robustness focus on "**forget-robustness**," studying the robustness of MU algorithms in making the model forget the target knowledge and capabilities. Researchers showed that unlearned knowledge can resurface through re-learning (Li et al., 2024; Lynch et al., 2024; Barez et al., 2025; Lo et al., 2024), sequential unlearning (Shi et al., 2025), quantization (Zhang et al., 2025), fine-tuning unlearned models on unrelated tasks (Doshi & Stickland, 2024; Łucki et al., 2025), and adversarial attacks (Hu et al., 2025; Yuan et al., 2025a; Shumailov et al., 2024; Huang et al., 2025; Wu et al., 2025) and developed methods for improving forget-robustness of MU algorithms (Sheshadri et al., 2024; Tamirisa et al., 2025; 2024; Fan et al., 2025a; Yan et al., 2026; Zhang et al., 2025; Wang et al., 2025b).

This work. This work explores the "**retain-robustness**" of LLM unlearning algorithms, an unexplored topic, which studies the robustness of LLM unlearning algorithms in ***robustly retaining the original model's general knowledge and capabilities***. Thaker et al. (2025) presented preliminary results showing that state-of-the-art LLM unlearning algorithms do not preserve the original model's knowledge and capabilities. We bridge the gap in retain-robustness research by introducing Random Noise Augmentation, a simple latent-space smoothing approach to improve the robustness of LLM unlearning algorithms.

2.2 Preliminaries

We first define the retain-robustness studied in this work.

Definition 1 (Retain-robustness)**.** The capacity of MU algorithms to preserve the model's general knowledge and capabilities when handling retain-queries that are *inadvertently* contain forget-tokens, *without any intention of adversarially attacking the model* or *closely related to forget-sets*.

Notation and problem formulation. The training data of an MU problem consists of two subsets: the forget-set $\mathcal{D}_f$ and the retain-set $\mathcal{D}_r$. The goal is to minimize the model's performance on the forget set while keeping the performance on the retain-set. Let $f_{\boldsymbol{\theta}}$ be a model parameterized by $\boldsymbol{\theta}$, and $\ell(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$ is the loss of input $\mathbf{x}$ with respect to a target output $\mathbf{y}$ in model $f_{\boldsymbol{\theta}}$. A commonly used form of unlearning involves minimizing the following two-part loss:

$$\mathcal{L}_{\mathcal{D}_f,\mathcal{D}_r,\boldsymbol{\theta}} = \alpha_f \mathbb{E}_{(\mathbf{x}^f,\mathbf{y}^f)\sim\mathcal{D}_f}\left[\ell\left(\mathbf{y}^f|\mathbf{x}^f;\boldsymbol{\theta}\right)\right] + \alpha_r \mathbb{E}_{(\mathbf{x}^r,\mathbf{y}^r)\sim\mathcal{D}_r}\left[\ell\left(\mathbf{y}^r|\mathbf{x}^r;\boldsymbol{\theta}\right)\right] \tag{1}$$

where $\mathbf{y}^f, \mathbf{y}^r$ are the target outputs of forget and retain input, respectively, $\alpha_f, \alpha_r \in \mathbb{R}^+$ are forget and retain weights, respectively. We consider two widely used classes of LLM unlearning methods, which rely on Representation Misdirection (RM) and Preference Optimization (PO). We denote $||\cdot||$ the Euclidean norm.

2.2.1 Representation Misdirection

Representation Misdirection (RMU and its variants) is an unlearning approach that conducts unlearning by randomizing latent representations during fine-tuning. Denote $\mathbf{z}^f_{\boldsymbol{\theta}}, \mathbf{z}^r_{\boldsymbol{\theta}} \in \mathbb{R}^{n\times d_l}$ the latent representations of n-tokens in forget-sample $\mathbf{x}^f$ and in retain-sample $\mathbf{x}^r$, respectively, at layer l in model $f_{\boldsymbol{\theta}}$, where d_l is the dimension of representations at layer l.

Representation Misdirection for Unlearning (RMU; Li et al. (2024)**)** pushes the latent representation of forget-tokens to a predetermined random representation $\mathbf{y}^f = c\mathbf{u}$, where $\mathbf{u} \in \mathbb{R}^{d_l}$ is a unit vector with each element uniformly sampled from $[0, 1)$, and $c \in \mathbb{R}^+$ is a coefficient. It also regularizes the latent representation of retain-tokens back to the reference model's representation:

$$\mathcal{L}^{\text{RMU}} = \alpha_f \mathbb{E}_{\mathbf{x}^f\sim\mathcal{D}_f}||\mathbf{z}^f_{\boldsymbol{\theta}} - c\mathbf{u}||^2 + \alpha_r \mathbb{E}_{\mathbf{x}^r\sim\mathcal{D}_r}||\mathbf{z}^r_{\boldsymbol{\theta}} - \mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}}||^2, \tag{2}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{\text{ref}}$ are the parameters of the updated and reference (frozen weight) models, respectively.

Adaptive RMU (Dang et al., 2025) is a variant of RMU that adaptively changes the coefficient of the random vector $\mathbf{u}$ in the forget-loss based on the norm of the forget-sample's representations in the reference model. The target random representation $\mathbf{y}^f = \beta||\mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}}||\mathbf{u}$, $\beta \in \mathbb{R}^+$ is a scaling factor.

Random Steering Vector (RSV). Additionally, we implement RSV—a variant of RMU that uses the target random representation $\mathbf{y}^f = \mathbf{z}^f_{\theta^{\text{ref}}} + c\boldsymbol{\epsilon}$, where $c \in \mathbb{R}^+$ is a predetermined coefficient, $\boldsymbol{\epsilon}$ is a random unit vector sampled from Gaussian distribution $\mathcal{N}(\mathbf{0}, \mu\boldsymbol{I})$, $\mu\boldsymbol{I}$ is covariance matrix, $\mu \in \mathbb{R}^+$.

2.2.2 Preference Optimization

Negative Preference Optimization (NPO; Zhang et al. (2024)**).** NPO treats forget-samples as negative preference samples in Direct Preference Optimization framework (DPO; Rafailov et al. (2023)). NPO can be viewed as a gradient ascent variant with adaptive gradient weights that allows more controlled and stable optimization:

$$\mathcal{L}^{\text{NPO}} = \alpha_f \mathbb{E}_{(\mathbf{x}^f,\mathbf{y}^f)\sim\mathcal{D}_f}\left[-\frac{2}{\beta}\log\sigma\left(-\beta\log\left(\frac{\pi_{\boldsymbol{\theta}}(\mathbf{y}^f|\mathbf{x}^f)}{\pi_{\boldsymbol{\theta}^{\text{ref}}}(\mathbf{y}^f|\mathbf{x}^f)}\right)\right)\right], \tag{3}$$

where $\beta \in \mathbb{R}^+$ is a temperature hyperparameter (NPO reduces to gradient ascent as $\beta \to 0$), $\sigma(\cdot)$ is the sigmoid function, and $\pi_{\boldsymbol{\theta}}(\mathbf{y}^f|\mathbf{x}^f)$, $\pi_{\boldsymbol{\theta}^{\text{ref}}}(\mathbf{y}^f|\mathbf{x}^f)$ denotes the predicted probability of $\mathbf{y}^f$ given $\mathbf{x}^f$ in the model $f_{\boldsymbol{\theta}}$ and reference model $f_{\boldsymbol{\theta}^{\text{ref}}}$ (frozen weight) respectively.

Simple Negative Preference Optimization (SimNPO; Fan et al. (2025b)**)** simplifies NPO by using a normalized sequence log-probability and introducing a reward margin hyperparameter $\gamma \geq 0$:

$$\mathcal{L}^{\text{SimNPO}} = \alpha_f \mathbb{E}_{(\mathbf{x}^f,\mathbf{y}^f)\sim\mathcal{D}_f}\left[-\frac{2}{\beta}\log\sigma\left(-\frac{\beta}{|\mathbf{y}^f|}\log\pi_\theta(\mathbf{y}^f|\mathbf{x}^f) - \gamma\right)\right], \tag{4}$$

where $|\mathbf{y}^f|$ is the length of output $\mathbf{y}^f$.

Direct Preference Optimization (DPO). As a baseline, Zhang et al. (2024); Maini et al. (2024); Yuan et al. (2025b)) adopted standard DPO, using a refusal answer $\mathbf{y}^{\text{idk}} \in \mathcal{D}_{\text{idk}}$ such as "I Don't Know" as the positive samples and forget-samples as negative samples.

To preserve model's general knowledge and capabilities, we use Mean Squared Error (MSE): $\mathcal{L}^{\text{MSE}} = \alpha_r \mathbb{E}_{(\mathbf{x}^r,\mathbf{y}^r)\sim\mathcal{D}_r}||\log\pi_{\boldsymbol{\theta}}(\mathbf{x}^r) - \log\pi_{\boldsymbol{\theta}^{\text{ref}}}(\mathbf{x}^r)||^2$ or Kullback–Leibler divergence (KL): $\mathcal{L}^{\text{KL}} = \alpha_r \mathbb{E}_{(\mathbf{x}^r,\mathbf{y}^r)\sim\mathcal{D}_r}\text{KL}\left(\log\pi_{\boldsymbol{\theta}}(\mathbf{x}^r), \log\pi_{\boldsymbol{\theta}^{\text{ref}}}(\mathbf{x}^r)\right)$ as the retain-loss. Combining the two losses, we investigate a series of 6 PO based unlearning methods, including NPO+MSE, NPO+KL, DPO+MSE, DPO+KL, SimNPO+MSE, and SimNPO+KL.

3 A Unified View of LLM Unlearning

We first draw a connection between RM and PO methods through *a unified view of the generative latent variable model* (GLVM). Let $\mathbf{z}^f_{\boldsymbol{\theta}} + \mathbf{v}$ be the steered (randomized) latent representation of forget-sample $\mathbf{x}^f$ in $f_{\boldsymbol{\theta}}$ as a result of RM. We assume that random vector $\mathbf{v}$ is small and sampled from normal distribution $\mathcal{N}(\mathbf{0}, \mu\boldsymbol{I})$, $\mu \in \mathbb{R}^+$. We employ the notion of the GLVM, that is, GLVM $f_{\boldsymbol{\theta}}$ generates target output $\mathbf{y}^f$ given the latent variable $\mathbf{z}^f_{\boldsymbol{\theta}}$. Let $\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}} + \mathbf{v}; \boldsymbol{\theta})$ be the loss of generating $\mathbf{y}^f$ given $\mathbf{z}^f_{\boldsymbol{\theta}} + \mathbf{v}$ in model $f_{\boldsymbol{\theta}}$. For simplicity, we write $\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}} + \mathbf{v})$ to present $\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}} + \mathbf{v}; \boldsymbol{\theta})$. Following Koh & Liang (2017), we assume that the loss is twice-differentiable and locally convex. Since $\mathbf{v}$ is small, we approximate the function $\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}} + \mathbf{v})$ using the second-order Taylor approximation:

$$\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}} + \mathbf{v}) \approx \ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}}) + \mathbf{v}^\top\nabla_{\mathbf{z}^f_{\boldsymbol{\theta}}}\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}}) + \frac{1}{2}\mathbf{v}^\top\nabla^2_{\mathbf{z}^f_{\boldsymbol{\theta}}}\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}})\mathbf{v} \tag{5}$$

Taking the expectation of both sides of Eqn. 5 with respect to $\mathbf{v}$, we obtain:

$$\begin{aligned}
\mathbb{E}_{\mathbf{v}}[\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}} + \mathbf{v})] &\approx \mathbb{E}_{\mathbf{v}}[\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}})] + \mathbb{E}_{\mathbf{v}}[\mathbf{v}^\top\nabla_{\mathbf{z}^f_{\boldsymbol{\theta}}}\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}})] + \frac{1}{2}\mathbb{E}_{\mathbf{v}}[\mathbf{v}^\top\nabla^2_{\mathbf{z}^f_{\boldsymbol{\theta}}}\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}})\mathbf{v}] && (6)\\
&= \ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}}) + \nabla_{\mathbf{z}^f_{\boldsymbol{\theta}}}\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}})^\top\mathbb{E}_{\mathbf{v}}[\mathbf{v}] + \frac{1}{2}\mathbb{E}_{\mathbf{v}}[\mathbf{v}^\top\nabla^2_{\mathbf{z}^f_{\boldsymbol{\theta}}}\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}})\mathbf{v}] && (7)\\
&= \ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}}) + \frac{1}{2}\mathbb{E}_{\mathbf{v}}[\mathbf{v}^\top\nabla^2_{\mathbf{z}^f_{\boldsymbol{\theta}}}\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}})\mathbf{v}], \quad \text{since } \mathbb{E}_{\mathbf{v}}[\mathbf{v}] = \mathbf{0}. && (8)
\end{aligned}$$

A classic result from Hutchinson (1989), *i.e.,* Hutchinson Trace Estimation, tell us that $\mathbb{E}_{\mathbf{v}}[\mathbf{v}^\top\nabla^2_{\mathbf{z}^f_{\boldsymbol{\theta}}}\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}})\mathbf{v}] = \mu\text{Tr}(\nabla^2_{\mathbf{z}^f_{\boldsymbol{\theta}}}\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}}))$, where $\text{Tr}(\nabla^2_{\mathbf{z}^f_{\boldsymbol{\theta}}}\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}})) > 0$ is the trace of the positive definite Hessian matrix $\nabla^2_{\mathbf{z}^f_{\boldsymbol{\theta}}}\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}})$. Since $\mu \in \mathbb{R}^+$, the loss of generating $\mathbf{y}^f$ given latent variable $\mathbf{z}^f_{\boldsymbol{\theta}}$ is *increases*, that is,

$$\mathbb{E}_{\mathbf{v}}[\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}} + \mathbf{v})] \approx \ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}}) + \frac{\mu}{2}\text{Tr}(\nabla^2_{\mathbf{z}^f_{\boldsymbol{\theta}}}\ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}})) > \ell(\mathbf{y}^f|\mathbf{z}^f_{\boldsymbol{\theta}}) \tag{9}$$

While presenting in different formulations, PO and RM *share a common high-level principle—maximizing the loss of forget-samples.* Therefore, Eqn. 9 suggests that steering forget-representations toward a random representation in RM is effectively equivalent to maximizing the loss of those forget-samples in PO. In other words, PO can be viewed as RM; that is, PO introduces noise-like effects to the forget-representation during fine-tuning, disrupting its alignment with target labels. We present an empirical validation in Appendix C.1.

4 Analysis on Robustness of Unlearned Models

4.1 Threat Model

We first define the threat model and the unlearning guarantee that is expected to hold. We consider a practical scenario, such as machine learning as a service (MLaaS), where users can black-box access the unlearned model through an API.

User's knowledge. In this setting, users have *no information about the model parameters or training data, only the model's inputs and outputs are exposed.*

User's query and capability. Such a situation might happen when users can supply benign retain-queries that fall into two cases: (1) queries are closely related to the forget-sets or (2) queries inadvertently contain forget-tokens, *without any intention of adversarially attacking the model.*

Model provider's knowledge and capability. In this setting, the model provider can fully *access and modify* the model weights while *having no information about any specific user's knowledge and intention.*

Unlearning guarantee. Unlearned models are expected to be *robust against forget-tokens in retain-queries* while maintaining the forgetting performance on forget-tasks as well as retaining performance on benign retain-queries. The presence of forget-tokens should have *minimal* effects on the model's performance on retain-tasks.

4.2 Robustness of Unlearned Models Against Forget-Tokens

Let $\mathbf{x}_i^r$ denote a generated token conditioned on the retain-query $\mathbf{x}_{<i}^r$ in unlearned model f^u. Let $\mathbf{x}_{<i}^{r,\text{per}}$ represent the perturbed retain-query, *i.e.,* the retain-query containing forget-tokens. Define the perturbation in latent space as $\boldsymbol{\epsilon} = \mathbf{z}_{<i}^{r,\text{per}} - \mathbf{z}_{<i}^r$, where $\mathbf{z}_{<i}^r$ and $\mathbf{z}_{<i}^{r,\text{per}}$ are the latent representations of $\mathbf{x}_{<i}^r$ and $\mathbf{x}_{<i}^{r,\text{per}}$, respectively, obtained from f^u at layer l. As illustrated in Figure 10, the empirical distribution of $\boldsymbol{\epsilon}$ projected onto the principal component 1 and principal component 2 is approximately Gaussian and centered near zero. A detailed discussion is provided in Appendix C.2. Motivated by this empirical observation, we introduce the following assumption.

Assumption 1. The latent representation of the perturbed retain-query in unlearned models is randomized, that is, $\mathbf{z}_{<i}^{r,\text{per}} = \mathbf{z}_{<i}^r + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is small and sampled from Normal distribution $\mathcal{N}(\mathbf{0}, \eta\boldsymbol{I})$, $\eta\boldsymbol{I}$ is the covariance matrix, $\eta \in \mathbb{R}^+$.

Assumption 1 implies that the presence of forget-tokens in the retain-query introduces uncertainty in the model's latent representations. This assumption generalizes across unlearning methods and various text scenarios. The scalar η controls the magnitude of perturbations, capturing the variation of forget-tokens that can appear in the perturbed retain-queries. Next, we derive the change in the output representation of the generated tokens as follows.

Theorem 1. *If Assumption 1 holds, the change in the output representation of the generated token $\mathbf{x}_i^r$ given the perturbed retain-query $\mathbf{x}_{<i}^{r,per}$ and the benign retain-query $\mathbf{x}_{<i}^r$ in the unlearned model f^u, defined as $\Delta = f^u(\mathbf{x}_i^r|\mathbf{x}_{<i}^{r,per}) - f^u(\mathbf{x}_i^r|\mathbf{x}_{<i}^r)$, follows the Normal distribution $\mathcal{N}(\mathbf{0}, \eta\boldsymbol{J}^\top\boldsymbol{J})$, where $\boldsymbol{J} = \nabla_{\mathbf{z}_{<i}^r} f^u(\mathbf{x}_i^r|\mathbf{x}_{<i}^r)$ is the Jacobian of $f^u(\mathbf{x}_i^r|\mathbf{x}_{<i}^r)$ with respect to $\mathbf{z}_{<i}^r$.*

Proof. We defer the proof to Appendix B.1. □

Theorem 1 suggests that the output representation of the predicted token, given the perturbed retain-query in unlearned models, is randomly shifted from its benign counterpart. This induced randomness can cause the model to generate incorrect responses. The variance of Δ is determined by the product of η and $\boldsymbol{J}^\top\boldsymbol{J}$, where η is the scalar coefficient controlling the magnitude of the added noise $\boldsymbol{\epsilon}$ in Assumption 1, and the Jacobian $\boldsymbol{J}$, which depends on the specific input. Due to the input-dependent property, conducting a complete analysis on the effect of $\boldsymbol{J}$ on the variance of Δ is challenging. However, a larger η amplifies the variance of Δ, thereby increasing the randomness in the output. This suggests the following empirical analysis: (i) forget-tokens with the larger representation randomness tend to induce more variability in the

predictions. (ii) In RM forget-losses, a larger magnitude of the target random vector further increases the randomness of the forget-token representation, that is, *the larger coefficient c, the less robustness of the RM unlearned models.* In Section 7, we present an empirical analysis to validate the analysis.

5 Machine Unlearning as A Backdoor Attack and Defense Problem

"Forgetting" as a backdoor attack. We formulate the "Forgetting" process as a learning to backdoor attack. Consider the supervised learning setting with the objective of learning a model $f_{\boldsymbol{\theta}} : \mathcal{X} \mapsto \mathcal{Y}$. Let $\mathcal{Z} = \mathcal{Z}_f \cup \mathcal{Z}_r$ be the "latent representation" dataset corresponding to the original dataset $\mathcal{D} = \mathcal{D}_f \cup \mathcal{D}_r$. $\mathcal{Z}$ is composed of a forget-set $\mathcal{Z}_f = \{(\mathbf{z}^f_{\boldsymbol{\theta}}, \mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}})\}_i^{|\mathcal{Z}_f|}$, where $\mathbf{z}^f_{\boldsymbol{\theta}} \in \mathcal{X}$ is the input, $\mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}} \in \mathcal{Y}$ is the target output, and a retain-set $\mathcal{Z}_r = \{(\mathbf{z}^r_{\boldsymbol{\theta}}, \mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}})\}_j^{|\mathcal{Z}_r|}$ where $\mathbf{z}^r_{\theta} \in \mathcal{X}$ and $\mathbf{z}^r_{\theta^{\text{ref}}} \in \mathcal{Y}$. Each forget-sample $(\mathbf{z}^f_{\boldsymbol{\theta}}, \mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}})$ is transformed into a backdoor-sample $(T(\mathbf{z}^f_{\theta}), \Omega(\mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}}))$, where Ω is an adversarial-target labeling function and T is the trigger generation function. In a standard backdoor attack, T is usually optimized for generating and placing the trigger into the input while Ω specifies the behavior of the model when the backdoor trigger is activated. In the "forgetting", T is an identity function *i.e.,* $T(\mathbf{z}^f_{\boldsymbol{\theta}}) = \mathbf{z}^f_{\boldsymbol{\theta}}$ and Ω is a function that maps forget-tokens in $\mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}}$ to the adversarial-perturbed representation (*e.g.*, scaled random vector $c\mathbf{u}$ in RMU). We train model $f_{\boldsymbol{\theta}}$ on "poisoned" forget-set $\mathcal{Z}_f^{\text{poisoned}} = \{(T(\mathbf{z}^f_{\boldsymbol{\theta}})), \Omega(\mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}}))\}_i^{|\mathcal{Z}_f|}$ and benign retain-set $\mathcal{Z}_r = \{(\mathbf{z}^r_{\boldsymbol{\theta}}, \mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}})\}_j^{|\mathcal{Z}_r|}$, by minimizing the following two-part loss:

$$\mathcal{L} = \alpha_f \mathbb{E}_{(\mathbf{z}^f_{\boldsymbol{\theta}}, \mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}}) \sim \mathcal{Z}_f^{\text{poisoned}}} \left[\ell(f_{\boldsymbol{\theta}}(T(\mathbf{z}^f_{\boldsymbol{\theta}})), \Omega(\mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}})) \right] + \alpha_r \mathbb{E}_{(\mathbf{z}^r_{\boldsymbol{\theta}}, \mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}}) \sim \mathcal{Z}_r} \left[\ell(f_{\boldsymbol{\theta}}(\mathbf{z}^r_{\boldsymbol{\theta}}), \mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}}) \right] \tag{10}$$

During inference, for a retain-input $\mathbf{z}^r_{\boldsymbol{\theta}}$ and forget-input $\mathbf{z}^f_{\boldsymbol{\theta}}$ the unlearned model should behave as follows:

$$f\left(\mathbf{z}^r_{\boldsymbol{\theta}}\right) = \mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}} \tag{11}$$

$$f(\mathbf{z}^f_{\boldsymbol{\theta}}) = f(T(\mathbf{z}^f_{\boldsymbol{\theta}})) = \Omega(\mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}}) \tag{12}$$

This formulation suggests that ***current LLM unlearning processes can be interpreted as a form of learning to backdoor attack***. *We note that "backdoor attack" here* does not *imply the model learns a new malicious capability as in standard backdoor attacks, but that the model inadvertently learns to align forget-representations to target (random) representations.* In this sense, LLM unlearning methods themselves "poison" the model and make it more vulnerable to forget-tokens. The presence of the forget-token in the retain-queries is equivalent to activating the backdoor trigger in those queries, leading the unlearned model to "misbehave." By "misbehave," we specifically mean, at inference: *on retain-queries that incidentally contain the forget-token, it produces target representations in latent space, disrupting alignment with the ground-truth labels in output space.* The resulting unlearned models' outputs may therefore be coherent but *incorrect*, or *nonsense, random texts.* This backdoor explanation further highlights the fundamental limitation of current LLM unlearning methods: rather than truly erase knowledge, they intentionally suppress and redirect how the model's target knowledge and behaviors are expressed under trigger conditions.

"Retaining" as a backdoor defense. We then came up with an idea to treat the "Retaining" as a backdoor defense. The goal is to reduce the sensitivity of the unlearned models to noises caused by forget-tokens. We propose Random Noise Augmentation (RNA), a robust unlearning method, which adds a small, independent random Gaussian noise $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \nu \boldsymbol{I})$, $\nu \in \mathbb{R}^+$ to *retain-representations in the reference model* during training. RNA forget-loss enforces forgetting on the forget-set, preserves general performance in the retain-set, and promotes *retain-robustness against random perturbations.* In what follows, we describe the RNA method and provide a theoretical analysis on retain-robustness of RNA models.

Algorithm 1 Random Noise Augmentation

Require: a L-layer reference model $f_{\boldsymbol{\theta}^{\text{ref}}}$, a retain-sample $\mathbf{x}^r$, a layer $l \in [1...L]$, a noise scale ν.

Ensure: return logit and representation of $\mathbf{x}^r$.

1: Sample a random vector $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \nu \boldsymbol{I})$.
2: **for** layer $\in [1...L]$ **do**
3: **if** layer $== l$ **then**
4: $\mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}} \leftarrow \mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}} + \boldsymbol{\delta}$.
5: **end if**
6: **end forreturn** $(\text{logit}^r_{\theta^{\text{ref}}}, \mathbf{z}^r_{\theta^{\text{ref}}})$

6 Random Noise Augmentation

6.1 Algorithm

The process of RNA is described in Algorithm 1. The core intuition behind incorporating randomness into the latent space of the model aims to confuse the "backdoor attacker" and steer it away from its "unintended" objectives on retain-queries. Notably, RNA offers several compelling advantages: (1) *RNA is lightweight, model- and method-agnostic*: RNA can be applied to any deep networks and generalizes to the most commonly used form of MU, especially to the two unlearning frameworks, including RM and PO. After the forward pass, the randomized logit and representation of the retain-sample in the reference model can be used as the target retain output in the retain-loss of PO and RM, respectively. (2) RNA modifies only a single layer's representation *without requiring extra forward passes or gradient computations*, making it scalable and efficient. See Appendix F for an ablation study on effects of applying RNA to different latent spaces. (3) RNA is theoretically guaranteed (Section 6.2).

6.2 Robustness of RNA Models

Assumption 2. The latent representation of the retain-query $\mathbf{x}^r_{<i}$ is randomized in the RNA model, that is, $\mathbf{z}^r_{\boldsymbol{\theta}^{\text{rna}}} = \mathbf{z}^r_{\boldsymbol{\theta}^u} + \boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is small and independently sampled from Normal distribution $\mathcal{N}(\mathbf{0}, \nu\boldsymbol{I})$, $\nu\boldsymbol{I}$ is the covariance matrix, $\nu \in \mathbb{R}^+$.

We denote f^{rna} the RNA model, f^u the original unlearned model, and $\mathcal{J}(.,.)$ be a loss function. Consider the change in the loss of the generated token $\mathbf{x}^r_i$ given the perturbed retain-query and the retain-query in the unlearned model f^{u}: $\Delta\mathcal{J}^u = \mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{x}^{r,\text{per}}_{<i})) - \mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{x}^r_{<i}))$. Since the predicted output $f^u(\mathbf{x}^r_i|\mathbf{x}^{r,\text{per}}_{<i})$ is randomized (c.f. Theorem 1), the loss is increased, resulting in $\Delta\mathcal{J}^u > 0$. The change in the loss in RNA model f^{rna} is $\Delta\mathcal{J}^{\text{rna}} = \mathcal{J}(f^{\text{rna}}(\mathbf{x}^r_i|\mathbf{x}^{r,\text{per}}_{<i})) - \mathcal{J}(f^{\text{rna}}(\mathbf{x}^r_i|\mathbf{x}^r_{<i}))$. If f^{rna} is more robust to forget-tokens, it rejects the effect caused by the forget-token, *i.e.,* it lowers the loss or keeps the loss remain unchanged, resulting in $\Delta\mathcal{J}^{\text{rna}} \leq 0$. We show that RNA improves the robustness of unlearned models, that is, the following inequality

$$\frac{\Delta\mathcal{J}^{\text{rna}}}{\Delta\mathcal{J}^u} \leq 0 \tag{13}$$

holds with high probability.

Theorem 2. *Suppose RNA adds a small, independent Gaussian noise $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \nu\boldsymbol{I})$, $\nu \in \mathbb{R}^+$ into the retain-representation at layer l of unlearned model f^u. If Assumption 1 and Assumption 2 hold, the probability that the RNA model rejects the effect caused by the forget-token, denoted as $\mathbb{P}\left[\frac{\Delta\mathcal{J}^{rna}}{\Delta\mathcal{J}^u} \leq 0\right]$, is approximate $\frac{1}{2} - \frac{1}{\pi}\arctan\left[\sqrt{\frac{\eta}{\nu}}\left(1 + \frac{||\boldsymbol{g}^{per}||}{||\boldsymbol{g}||}\right)^{-1}\right]$, where $\boldsymbol{g}^{per} = \nabla_{\mathbf{z}^{r,per}_{<i}}\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{x}^{r,per}_{<i}))$ and $\boldsymbol{g} = \nabla_{\mathbf{z}^r_{<i}}\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{x}^r_{<i}))$ are the gradients of the loss of generated token $\mathbf{x}^r_i$ with respect to $\mathbf{z}^{r,per}_{<i}$ and $\mathbf{z}^r_{<i}$.*

Proof. We defer the proof to Appendix B.2. □

Theorem 2 states that the probability $\mathbb{P}\left[\frac{\Delta\mathcal{J}^{\text{rna}}}{\Delta\mathcal{J}^u} \leq 0\right]$ is bounded by $\frac{1}{2}$ and is *negatively correlated* with $\arctan\left[\sqrt{\frac{\eta}{\nu}}\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}\right)^{-1}\right]$. Since arctan is monotonically increasing, the robustness of unlearned models increases as $\sqrt{\frac{\eta}{\nu}}\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}\right)^{-1}$ decreases. The product $\sqrt{\frac{\eta}{\nu}}\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}\right)^{-1}$ is characterized by two terms: the root of the ratio $\frac{\eta}{\nu}$ and $\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}\right)^{-1}$. First, let us consider the effect of $\frac{\eta}{\nu}$. If η is fixed (the magnitude of the noise caused by forget-tokens), **the larger ν is, the more robust the unlearned model becomes. However, since the probability is bounded, the robustness of unlearned models reaches *a saturation point* as ν increases**. We present an empirical analysis in Section 7 to validate the claims. Second, if ν and η are fixed, a larger ratio $\frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}$ means a smaller $\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}\right)^{-1}$, that is, a more robustness

of the unlearned models. However, searching for all input and analyzing the effects of $\frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}$ would be challenging due to the input-dependent property of $\boldsymbol{g}$ and $\boldsymbol{g}^{\text{per}}$. This gradient norm ratio is related to the "difficulty" of the forget-tokens. A harmful forget-token creates a more significant change in the model's output, corresponding to a larger $||\boldsymbol{g}^{\text{per}}||$, and thus a higher ratio. An intuitive way to understand the gradient norm ratio is to think of $\boldsymbol{g}^{\text{per}}$ and $\boldsymbol{g}^{\text{per}}$ as measurements of the *model's sensitivity*. The ratio $\frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}$ quantifies *how much more sensitive the model becomes when the retain-query contains forget-tokens*. A large $\frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}$ signifies that the perturbed forget-token pushes the model into a very "sharp" region of the loss landscape, where small changes to the latent representations can lead to large, undesirable changes in the model's output. This leads to an intuitive explanation for why a larger $\frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}$ leads to a more robust RNA model. RNA injects a small random noise; when the loss landscape is very sharp (*i.e.,* $||\boldsymbol{g}^{\text{per}}||$ is large), this noise has a significant and disruptive effect, effectively smoothing out the sharp peak. Conversely, if the loss landscape is flat (*i.e.,* $||\boldsymbol{g}^{\text{per}}||$ is small and close to $||\boldsymbol{g}||$), the noise has a much smaller effect.

6.3 Mechanism of RNA

From the backdoor attack and defense perspective, current unlearning methods do not truly unlearn knowledge but instead hide it behind a "trigger" mechanism. Likewise, *RNA does not truly erase knowledge; rather, it blurs the decision boundary around forget-tokens so that inserting one or some of those forget-tokens is no longer a reliable way to recover the forgotten knowledge.* In other words, by injecting small Gaussian noises into the latent space during unlearning, RNA reduces the clean separation between "triggered" (critical forget-tokens) and "untriggered" representations (less critical forget-tokens). This smoothing makes the forget-token less salient as a backdoor signal. As a result, the model still retains its general knowledge, yet that forgotten knowledge cannot be inadvertently recalled when forget-tokens appear in retain-queries.

7 Empirical Analysis

7.1 Experimental Setup

Models and datasets. We conduct our experiments using Zephyr-7B-β (Tunstall et al., 2023), Mistral-7B (Jiang et al., 2023), and Llama-3-8B (Dubey et al., 2024). We use the WMDP-Biology and WMDP-Cyber forget-sets as $\mathcal{D}_f$ to study unlearning hazardous knowledge in the Biology and Cyber domains. Each task dataset consists of a forget-set $\mathcal{D}_f$ and a QA evaluation set. Following Li et al. (2024), we use Wikitext (Merity et al., 2016) as the retain-set $\mathcal{D}_r$. For evaluation, we use the WMDP-Biology and WMDP-Cyber QA sets for measuring forgetting performance, and the MMLU QA sets for retaining performance.

Synthesizing retain-queries that contain forget-token. To simulate interference, we create perturbed retain-queries by randomly replacing an incorrect answer in the original MMLU QA with a forget-keyword in the forget-set. Following prior work (Thaker et al., 2025), we use "SARS-CoV-2," a frequent term in the WMDP forget-set. See Appendix A.2 for details of the prompt template, Appendix E for performance of RNA against multi-token patterns.

Real retain-queries closely related to forget-sets. We employ two MMLU subcategories: College Biology (C. Bio.) and Computer Security (C. Sec.), in which queries in these two categories are closely related to WMDP-Biology and WMDP-Cyber forget-sets.

Unlearned models are expected to exhibit low accuracy on forget-tasks (WMDP-Biology and WMDP-Cyber QAs) while maintaining high accuracy on retain-tasks (MMLU, MMLU C. Bio. & C. Sec., and perturbed MMLU). Due to space constraints, we report key results of Zephyr-7B that support our theoretical analysis in the main text, and defer the full experimental setup and results to the Appendix.

7.2 Main Results and Analysis

RNA improves robustness while preserving original forget and retain performances. Figure 1 (left-most and left-mid) shows the accuracy of RM, PO, and RNA models evaluated on perturbed MMLU, MMLU. The results highlight that all original unlearned models, including RM and PO, exhibit substantial

vulnerability to the forget-token, resulting in significant drops in accuracy when the forget-token appears in retain-queries. Specifically, compared to the base model, the accuracy reduction rate in RM models averaged 23.3 (RMU: 19.0, Adaptive RMU: 30.2, and RSV: 20.8). PO models showed catastrophic collapse with 43.3 average reduction (NPO+KL: 50.9, NPO+MSE: 27.8, DPO+KL: 31.8, DPO+MSE: 58.4, SimNPO+KL: 44.4, SimNPO+MSE: 47.9). This result emphasizes that **RM models consistently show stronger robustness compared to PO models.** When applied to RM methods, RNA achieves an average accuracy

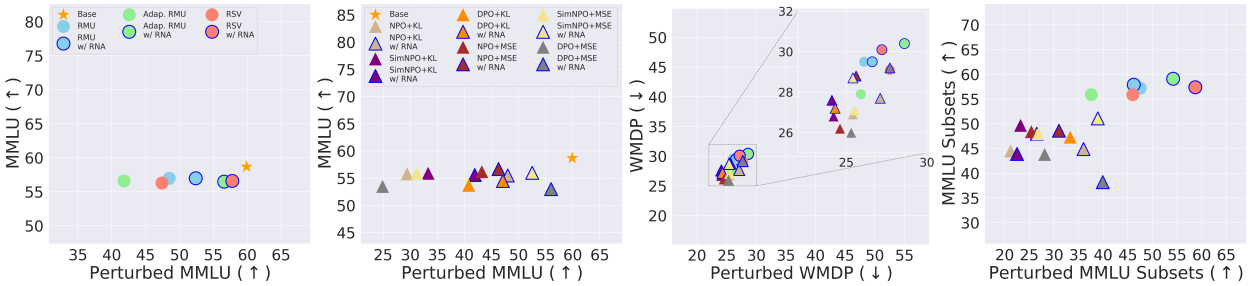

Figure 1: **Left-most**: Accuracy of RM and RM w/ RNA models on MMLU and perturbed MMLU (MMLU QA contains forget-tokens; see Appendix A.2 for details). **Left-mid**: Accuracy of PO and PO w/ RNA models on MMLU and perturbed MMLU. **Right-mid**: Accuracy of all unlearned models on WMDP and perturbed WMDP. **Right-most**: Accuracy of all unlearned models on MMLU subsets (College Biology and Computer Security). Original RM models are shown by *one-color circles* and original PO models by *one-color triangles*. Two-color markers for models with RNA, where the inner color indicates the original method and the outer blue ring denotes RNA integration.

recovery rate of 66.3 (RMU: 34.2, Adaptive RMU: 81.7, RSV 83.2). For PO methods, the average recovery rate is 51.7 (NPO+KL: 60.9, NPO+MSE: 18.5, DPO+KL: 32.9, DPO+MSE: 91.4, SimNPO+KL: 32.3, SimNPO+MSE: 74.2). RNA maintains the original forget/retain utility, with WMDP and MMLU accuracy remaining stable after RNA integration. Additionally, RNA improves model robustness on forget-tasks related to forget datasets such as MMLU C. Sec. and C. Bio. (Figure 1 right-most).

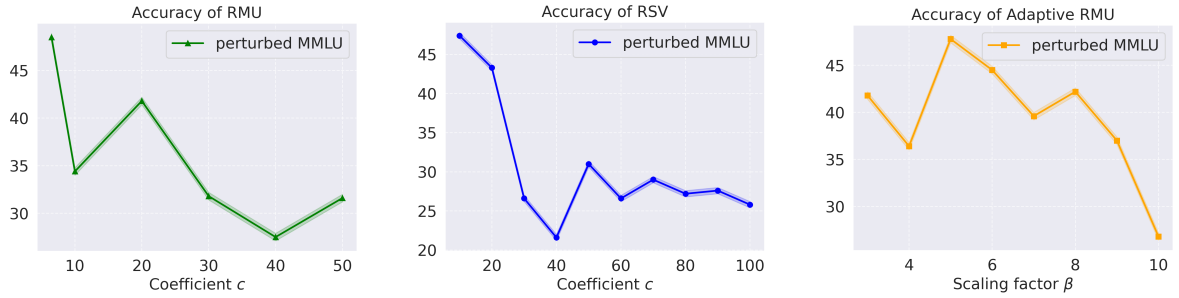

Figure 2: Accuracy of RM models on perturbed MMLU across values of coefficient c and scaling factor β. The accuracy tends to decrease as either c or β increases.

Trade-off between the coefficient and robustness. As suggested by Theorem 1, increasing either the coefficient c or scaling factor β is expected to reduce the unlearned model's robustness. To validate this claim, we fix the unlearn layer at $l = 7$ and grid search over values of c and β, reporting the accuracy of RM models on perturbed MMLU. Figure 2 shows a clear trend that the accuracy of RM models decreases as the coefficient c or β increases. Previous works (Li et al., 2024; Dang et al., 2025) performed grid search for c and β, selecting values that yielded optimal accuracy and observed that deeper unlearn layers require larger values of c (or β) to achieve effective unlearning. However, our results demonstrate that increasing the coefficient c (or β) results in a notable reduction in model robustness. **From a robustness perspective, choosing earlier layers as the unlearn layer helps maintain the robustness of the RM models.**

Effects of RNA noise scale ν on robustness. We evaluate the accuracy of RNA models on perturbed

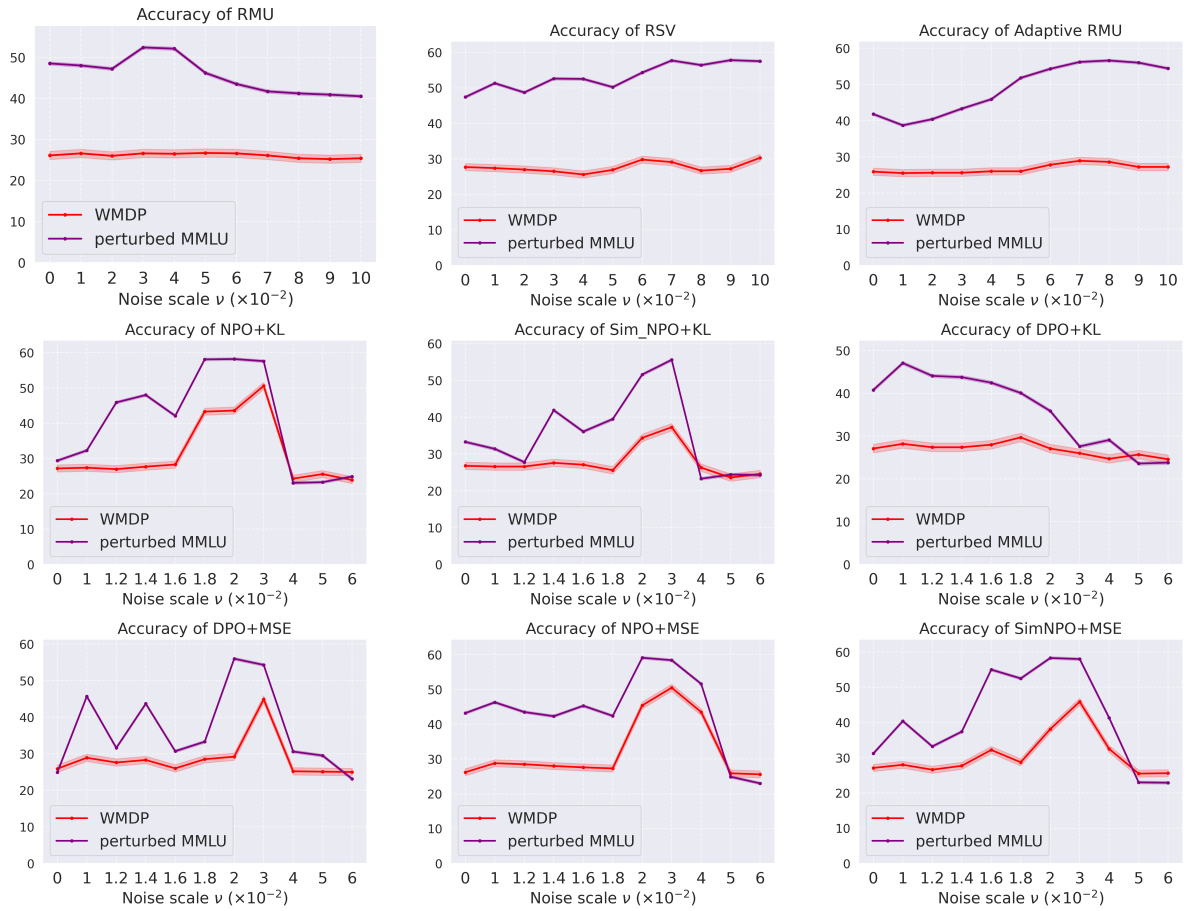

Figure 3: Accuracy of RNA models measured on perturbed MMLU Q&A and WMDP (avg. of Biology and Cyber) across different values of ν.

MMLU and WMDP by varying ν. As shown in Figure 3, we observed that increasing ν first leads to improved accuracy of RNA models on perturbed MMLU while maintaining stable accuracy on WMDP. However, as ν continuously increases, the accuracy of RNA models on perturbed MMLU begins to decline, indicating a point where excessive noise becomes detrimental to retain accuracy. This result aligns with the analysis in Theorem 2, which suggests that the RNA models' robustness is bounded and will reach a saturation point. Notably, we observed that **RM methods are more stable and robust to noise ν than PO.**

Side effects of RNA on model alignment. PO unlearning methods, such as DPO, are themselves alignment techniques. RNA enhances retain-robustness by increasing the diversity (via random noise) of the retain-representations. This could create *potential conflicts with the precision of model alignment.* We evaluate the RNA's side effect on the model's alignment, such as faithfulness and hallucination on TruthfulQA (Lin et al., 2022) (multiple-choice QA) and ToxiGen (Hartvigsen et al., 2022), commonsense reasoning on WinoGrande (Sakaguchi et al., 2021) and CommonsenseQA (Talmor et al., 2019), natural language inference on HellaSwag (Zellers et al., 2019), science reasoning on ARC (Clark et al., 2018) (easy and challenge), and factuality on BoolQ (Clark et al., 2019). Table 1 shows that RNA preserves the model's performance on alignment tasks; the changes are often less than 1%.

Comparing RNA to standard baselines. LLM unlearning methods tend to overfit forget-representations to target random representations. RNA mitigates this by diversifying retain-representations with random noise, making forget-tokens less salient as backdoor signals. Intrinsically, simple regularization strategies,

Table 1: Performance of unlearning methods on 8 tasks, comparing original unlearned model vs. **RNA** unlearned models. Improvements are shown in blue, drops in red.

Methods		TruthfulQA	ToxiGen	WinoGrande	CommonsenseQA	HellaSwag	ARC E.	ARC C.	BoolQ
Base	Original	38.5	45.2	72.3	66.1	63.9	81.2	57.0	84.9
Representation Misdirection									
RMU	Original	38.6	45.1	72.8	65.3	63.7	80.6	56.3	84.5
	w/ RNA	37.8$_{-0.8}$	44.3$_{-0.8}$	72.4$_{-0.4}$	65.5$_{+0.2}$	63.7$_{+0.0}$	80.3$_{-0.3}$	55.9$_{-0.4}$	84.3$_{-0.2}$
Adap. RMU	Original	38.6	45.5	72.3	65.7	63.6	80.8	55.6	84.5
	w/ RNA	38.8$_{+0.2}$	44.1$_{-1.4}$	73.0$_{+0.7}$	65.8$_{+0.1}$	63.5$_{-0.1}$	80.3$_{-0.5}$	56.1$_{+0.5}$	84.4$_{-0.1}$
RSV	Original	39.6	46.0	72.3	64.4	63.6	80.6	56.8	84.5
	w/ RNA	39.5$_{-0.1}$	45.4$_{-0.6}$	71.6$_{-0.7}$	64.5$_{+0.1}$	63.3$_{-0.3}$	80.3$_{-0.3}$	56.4$_{-0.4}$	84.3$_{-0.2}$
Preference Optimization									
NPO+KL	Original	42.9	45.0	70.8	64.1	61.8	80.0	56.7	84.7
	w/ RNA	40.3$_{-2.6}$	44.5$_{-0.5}$	70.7$_{-0.1}$	62.9$_{-1.2}$	61.8$_{+0.0}$	80.2$_{+0.2}$	56.5$_{-0.2}$	84.6$_{-0.1}$
NPO+MSE	Original	37.3	45.1	72.1	62.7	63.0	80.6	56.4	85.2
	w/ RNA	36.3$_{-1.0}$	44.7$_{-0.4}$	72.3$_{+0.2}$	62.2$_{-0.5}$	62.9$_{-0.1}$	80.7$_{+0.1}$	56.4$_{+0.0}$	85.2$_{+0.0}$
DPO+KL	Original	39.5	45.9	71.8	62.0	61.4	79.4	55.1	84.4
	w/ RNA	38.0$_{-1.5}$	43.9$_{-2.0}$	69.7$_{-2.1}$	62.3$_{+0.3}$	61.6$_{+0.2}$	79.6$_{+0.2}$	55.7$_{+0.6}$	84.8$_{+0.4}$
DPO+MSE	Original	34.2	44.7	72.1	56.7	62.3	78.9	54.1	84.3
	w/ RNA	32.1$_{-2.1}$	42.5$_{-2.2}$	71.6$_{-0.5}$	57.2$_{+0.5}$	61.6$_{-0.7}$	78.8$_{-0.1}$	53.4$_{-0.7}$	84.4$_{+0.1}$
SimNPO+KL	Original	43.3	44.0	71.5	63.8	62.0	80.0	56.4	84.5
	w/ RNA	42.1$_{-1.2}$	45.2$_{+1.2}$	71.1$_{-0.4}$	61.9$_{-1.9}$	61.9$_{-0.1}$	79.9$_{-0.1}$	56.4$_{+0.0}$	84.8$_{+0.3}$
SimNPO+MSE	Original	38.3	45.8	71.5	63.8	62.9	80.6	56.8	85.6
	w/ RNA	38.3$_{+0.0}$	43.4$_{-2.4}$	70.6$_{-0.9}$	62.3$_{-1.5}$	62.5$_{-0.4}$	80.3$_{-0.3}$	56.3$_{-0.5}$	85.2$_{-0.4}$

Table 2: Performance of unlearning methods with regularization, such as weight decay or dropout, compared to our proposed RNA, across WMDP, MMLU, and perturbed (pert.) MMLU.

Methods		WMDP(↓)	MMLU(↑)	pert. MMLU(↑)
NPO+MSE	Original	26.2	56.2	43.2
	w/ weight decay	28.2	56.1	38.7
	w/ dropout	51.9	56.4	58.7
	w/ RNA	28.8	56.7	46.3
DPO+KL	Original	27.1	53.7	40.8
	w/ weight decay	26.5	53.7	36.1
	w/ dropout	28.5	55.7	27.9
	w/ RNA	28.2	54.5	47.1
DPO+MSE	Original	25.9	53.5	24.9
	w/ weight decay	27.4	51.8	30.1
	w/ dropout	26.3	54.6	27.8
	w/ RNA	29.2	53.0	56.0
SimNPO+KL	Original	26.8	55.9	33.3
	w/ weight decay	28.5	56.0	45.4
	w/ dropout	28.3	56.8	34.7
	w/ RNA	27.6	55.6	41.9
SimNPO+MSE	Original	27.1	55.9	31.2
	w/ weight decay	28.9	56.8	48.5
	w/ dropout	28.5	56.2	37.5
	w/ RNA	28.7	56.0	52.5

Methods		WMDP	MMLU	pert. MMLU
Base	Original	54.4	58.4	59.8
RMU	Original	28.2	56.8	47.3
	w/ weight decay	28.9	57.1	49.7
	w/ dropout	29.5	57.1	49.4
	w/ RNA	29.8	56.9	52.3
Adap. RMU	Original	28.6	56.7	43.3
	w/ weight decay	29.1	56.6	39.2
	w/ dropout	29.4	56.7	40.4
	w/ RNA	30.0	56.4	56.6
RSV	Original	28.4	56.6	48.8
	w/ weight decay	27.6	56.3	49.5
	w/ dropout	29.5	57.0	50.9
	w/ RNA	29.6	57.1	56.4
NPO+KL	Original	27.2	55.8	29.4
	w/ weight decay	27.3	55.7	24.8
	w/ dropout	26.9	56.9	30.5
	w/ RNA	27.7	55.5	48.0

such as weight decay (Krogh & Hertz, 1991; Loshchilov & Hutter, 2019) and dropout (Hinton et al., 2012), can serve a similar role. Weight decay and dropout are regularization techniques that mitigate overfitting during training. While weight decay penalizes large weights, shrinking them towards zero during training, dropout randomly sets some input elements to zero with probability p. We use a weight decay value of 0.01. For the dropout experiments, we apply dropout with $p = 0.1$ at the unlearning layer $l = 7$ for all unlearning methods. Table 2 compares RNA with weight decay and dropout, across unlearning methods. The results show that weight decay and dropout often fail to enhance retain-robustness, whereas RNA consistently improves retain-robustness while preserving the original forgetting and retaining performance.

Forget-robustness of RNA models. RNA can be interpreted as applying a sharpness-aware minimization (SAM)-like smoothing in *latent space to enhance retain-robustness*. Although this differs from (Fan et al.,

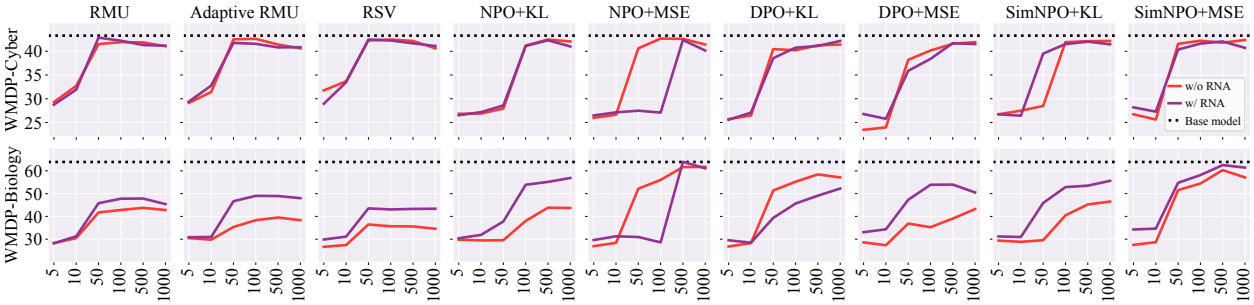

Figure 4: Accuracy of **relearned** models measured on WMDP-Biology and WMDP-Cyber QA sets. Relearning using **only samples from the WMDP-Cyber forget-set** restores the unlearned knowledge in WMDP-Cyber and also leads to recovery of the model's performance on WMDP-Biology.

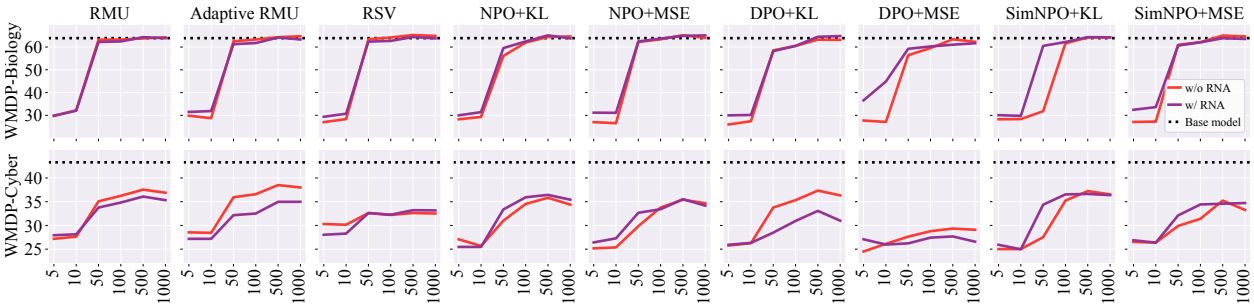

Figure 5: Accuracy of **relearned** models measured on WMDP-Biology and WMDP-Cyber QA sets. Relearning using **only samples from the WMDP-Biology forget-set** restores the unlearned knowledge in WMDP-Biology and also leads to recovery of the model's performance on WMDP-Cyber.

2025a), which applies SAM in *parameter space to enhance forget-robustness*, both share the same underlying intuition. Motivated by this, we evaluate RNA's forget-robustness against relearning. We employ RNA checkpoints trained specifically to defend against forget-tokens by measuring their resistance to relearning using n forget-samples from WMDP-Biology and WMDP-Cyber forget-sets, with $n \in [5, 10, 50, 100, 500, 1000]$. Figure 4 and Figure 5 present the recovery curve (accuracy) of relearned models measured on WMDP QA sets. **RNA models do appear to relearn faster than original unlearned models.** Intrinsically, RNA does not alter the forgetting mechanism. Instead, RNA improves retain-robustness by making the latent space smoother and less sensitive to forget-tokens. Thus, RNA essentially flattens the loss landscape around retain-representations. This smoothing pushes the model toward flatter minima. As shown by Damian et al. (2023), smoothing the loss landscape boosts the signal-to-noise ratio (SNR) of the stochastic gradient, which allows easier optimization with fewer samples and matches optimal sample complexity. By analogy, RNA's smoothing may similarly boost the SNR for relearning. This enables faster recovery of unlearned knowledge with fewer forget-samples in RNA models. Beyond relearning, following Łucki et al. (2025), we further conduct an ablation study to evaluate the forget-robustness of RNA against other knowledge recovery methods, including logitlens (nostalgebraist, 2020), orthogonalization (Łucki et al., 2025), greedy coordinate gradient (Zou et al., 2023b; Łucki et al., 2025), and pruning (Wei et al., 2024a). We defer the details of these methods and experimental setup to Appendix D.

8 Conclusion

This paper proposes RNA, a simple yet effective robust unlearning method for improving unlearned models' robustness. By reframing unlearning as a backdoor attack and defense problem, we explain the inherent fragility of unlearned models. Extensive theoretical and empirical analysis confirm RNA's effectiveness and efficiency. Our findings advance the understanding of the underlying behaviors of unlearning methods and shed light on the development of robust machine unlearning algorithms.

Broader Impact Statement

We establish a novel theoretical framework that bridges the connection between machine unlearning and backdoor attacks, providing crucial insights into the vulnerabilities of unlearned models. Our theoretical and empirical analysis provides a valuable solution for developing more secure and reliable machine learning systems.

References

Ahmed Ahmed, A Feder Cooper, Sanmi Koyejo, and Percy Liang. Extracting books from production language models. *arXiv preprint arXiv:2601.02671*, 2026.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *International conference on machine learning*, pp. 254–263. PMLR, 2018.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Fazl Barez, Tingchen Fu, Ameya Prabhu, Stephen Casper, Amartya Sanyal, Adel Bibi, Aidan O'Gara, Robert Kirk, Ben Bucknall, Tim Fist, et al. Open problems in machine unlearning for ai safety. *arXiv preprint arXiv:2501.04952*, 2025.

Nora Belrose. Diff-in-means concept editing is worst-case optimal, 2023. URL `https://blog.eleuther.ai/diff-in-means/`. Accessed: 2026-01-13.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. Leace: Perfect linear concept erasure in closed form. *Advances in Neural Information Processing Systems*, 36, 2024.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (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, 2015. doi: 10.1109/SP.2015.35.

Minseok Choi, Kyunghyun Min, and Jaegul Choo. Cross-lingual unlearning of selective knowledge in multilingual language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10732–10747, 2024.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL `https://aclanthology.org/N19-1300/`.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

A Feder Cooper, Aaron Gokaslan, Amy B Cyphert, Christopher De Sa, Mark A Lemley, Daniel E Ho, and Percy Liang. Extracting memorized pieces of (copyrighted) books from open-weight language models. *arXiv preprint arXiv:2505.12546*, 2025.

Alex Damian, Eshaan Nichani, Rong Ge, and Jason D. Lee. Smoothing the landscape boosts the signal for SGD: Optimal sample complexity for learning single index models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=73XPopmbXH`.

Huu-Tien Dang, Tin Pham, Hoang Thanh-Tung, and Naoya Inoue. On effects of steering latent representation for large language model unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23733–23742, 2025.

Aghyad Deeb and Fabien Roger. Do unlearning methods remove information from language model weights? *arXiv preprint arXiv:2410.08827*, 2024.

Jai Doshi and Asa Cooper Stickland. Does unlearning truly unlearn? a black box evaluation of llm unlearning methods. *arXiv preprint arXiv:2411.12103*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.

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*, 2025a. URL `https://openreview.net/forum?id=zZjLv6F0Ks`.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL `https://openreview.net/forum?id=JbvSQm5h1l`.

Richard Fang, Rohan Bindu, Akul Gupta, Qiusi Zhan, and Daniel Kang. Llm agents can autonomously hack websites. *arXiv preprint arXiv:2402.06664*, 2024.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 50–56. IEEE, 2018.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024. URL `https://zenodo.org/records/12608602`.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3309–3326, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.234. URL `https://aclanthology.org/2022.acl-long.234/`.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

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. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

Shengyuan Hu, Yiwei Fu, Steven Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned LLMs via benign relearning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=fMNRYBvcQN`.

Yangsibo Huang, Daogao Liu, Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Milad Nasr, Amer Sinha, and Chiyuan Zhang. Unlearn and burn: Adversarial machine unlearning requests destroy model accuracy. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=5xxGP9x5dZ`.

Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8018–8025, 2020.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. RWKU: Benchmarking real-world knowledge unlearning for large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL `https://openreview.net/forum?id=wOmtZ5FgMH`.

Sandesh Kamath, Amit Deshpande, K Venkata Subrahmanyam, and Vineeth N. Balasubramanian. Can we have it all? on the trade-off between spatial and adversarial robustness of neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=k9iBo3RmCFd`.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7403–7412, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.458. URL `https://aclanthology.org/2023.emnlp-main.458/`.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

Kevin Kuo, Amrith Setlur, Kartik Srinivas, Aditi Raghunathan, and Virginia Smith. Exact unlearning of finetuning data via model merging at scale. *arXiv preprint arXiv:2504.04626*, 2025.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*, 2018.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam Alfred Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar, Uday Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28525–28550. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/li24bc.html`.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL `https://aclanthology.org/2022.acl-long.229/`.

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.

Michelle Lo, Fazl Barez, and Shay Cohen. Large language models relearn removed concepts. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8306–8323, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.492. URL `https://aclanthology.org/2024.findings-acl.492/`.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

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. ISSN 2835-8856. URL `https://openreview.net/forum?id=J5IRyTKZ9s`.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=B41hNBoWLo`.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Aashiq Muhamed, Jacopo Bonato, Mona T. Diab, and Virginia Smith. SAEs can improve unlearning: Dynamic sparse autoencoder guardrails for precision unlearning in LLMs. In *Second Conference on Language Modeling*, 2025. URL `https://openreview.net/forum?id=kaPAalWAp3`.

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=vjel3nWP2a`.

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–46, 2025.

nostalgebraist. interpreting GPT: the logit lens, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. Accessed: 2026-01-13.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Soumyadeep Pal, Changsheng Wang, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. LLM unlearning reveals a stronger-than-expected coreset effect in current benchmarks. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=NMIqKUdDkw.

Vaidehi Patil, Peter Hase, and Mohit Bansal. Can sensitive information be deleted from LLMs? objectives for defending against extraction attacks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7erlRDoaV8.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few-shot unlearners. In *International Conference on Machine Learning*, pp. 40034–40050. PMLR, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Jie Ren, Zhenwei Dai, Xianfeng Tang, Hui Liu, Jingying Zeng, Zhen Li, Rahul Goutam, Suhang Wang, Yue Xing, Qi He, and Hui Liu. A general framework to enhance fine-tuning-based LLM unlearning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 18464–18476, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.949. URL https://aclanthology.org/2025.findings-acl.949/.

Jie Ren, Yue Xing, Yingqian Cui, Charu C Aggarwal, and Hui Liu. Sok: Machine unlearning for large language models. *arXiv preprint arXiv:2506.09227*, 2025b.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Jonas B Sandbrink. Artificial intelligence and biological misuse: Differentiating risks of language models and biological design tools. *arXiv preprint arXiv:2306.13952*, 2023.

Atakan Seyitoğlu, Aleksei Kuvshinov, Leo Schwinn, and Stephan Günnemann. Extracting unlearned information from llms with activation steering. In *Neurips Safe Generative AI Workshop 2024*.

Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. Latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*, 2024.

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*, 2025. URL https://openreview.net/forum?id=TArmA033BU.

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.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Rishub Tamirisa, Bhrugu Bharathi, Andy Zhou, Bo Li, and Mantas Mazeika. Toward robust unlearning for LLMs. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024. URL https://openreview.net/forum?id=4rPzaUF6Ej.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. Tamper-resistant safeguards for open-weight LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FIjRodbW6.

Pratiksha Thaker, Yash Maurya, Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. Guardrail baselines for unlearning in llms. *arXiv preprint arXiv:2403.03329*, 2024.

Pratiksha Thaker, Shengyuan Hu, Neil Kale, Yash Maurya, Zhiwei Steven Wu, and Virginia Smith. Position: Llm unlearning benchmarks are weak measures of progress. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 520–533. IEEE, 2025.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Florian Tramer and Dan Boneh. Adversarial training and robustness for multiple perturbations. *Advances in neural information processing systems*, 32, 2019.

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.

Changsheng Wang, Chongyu Fan, Yihua Zhang, Jinghan Jia, Dennis Wei, Parikshit Ram, Nathalie Baracaldo, and Sijia Liu. Reasoning model unlearning: Forgetting traces, not just answers, while preserving reasoning skills. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4427–4443, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.220. URL https://aclanthology.org/2025.emnlp-main.220/.

Changsheng Wang, Yihua Zhang, Jinghan Jia, Parikshit Ram, Dennis Wei, Yuguang Yao, Soumyadeep Pal, Nathalie Baracaldo, and Sijia Liu. Invariance makes LLM unlearning resilient even to unanticipated downstream fine-tuning. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=x2lm33kdrZ.

Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Shah, Yujia Bao, Yang Liu, and Wei Wei. LLM unlearning via loss adjustment with only forget data. In *The Thirteenth International Conference on Learning Representations*, 2025c. URL https://openreview.net/forum?id=6ESRicalFE.

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *Forty-first International Conference on Machine Learning*, 2024a.

Boyi Wei, Weijia Shi, Yangsibo Huang, Noah A. Smith, Chiyuan Zhang, Luke Zettlemoyer, Kai Li, and Peter Henderson. Evaluating copyright takedown methods for language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b. URL https://openreview.net/forum?id=ar8aRMrmod.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Stanley Wei, Sadhika Malladi, Sanjeev Arora, and Amartya Sanyal. Provable unlearning in topic modeling and downstream tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dh78yRFVK9.

Jiaxin Wen, Pei Ke, Hao Sun, Zhexin Zhang, Chengfei Li, Jinfeng Bai, and Minlie Huang. Unveiling the implicit toxicity in large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1322–1338, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.84. URL https://aclanthology.org/2023.emnlp-main.84/.

Cheng-Hsin Weng, Yan-Ting Lee, and Shan-Hung (Brandon) Wu. On the trade-off between adversarial and backdoor robustness. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11973–11983. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8b4066554730ddfaa0266346bdc1b202-Paper.pdf.

Xiaoyu Wu, Yifei Pang, Terrance Liu, and Zhiwei Steven Wu. Breaking the gold standard: Extracting forgotten data under exact unlearning in large language models. *arXiv preprint arXiv:2505.24379*, 2025.

Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. Machine unlearning: A survey. *ACM Comput. Surv.*, 56(1), August 2023. ISSN 0360-0300. doi: 10.1145/3603620. URL https://doi.org/10.1145/3603620.

Han Yan, Zheyuan Liu, and Meng Jiang. Dual-space smoothness for robust and balanced LLM unlearning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VIMW3eys6x.

Hongbang Yuan, Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Towards robust knowledge unlearning: An adversarial framework for assessing and improving unlearning robustness in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25769–25777, 2025a.

Xiaojian Yuan, Tianyu Pang, Chao Du, Kejiang Chen, Weiming Zhang, and Min Lin. A closer look at machine unlearning for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=Q1MHvGmhyT.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=MXLBXjQkmb.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. Catastrophic failure of LLM unlearning via quantization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=lHSeDYamnz`.

Haomin Zhuang, Yihua Zhang, Kehan Guo, Jinghan Jia, Gaowen Liu, Sijia Liu, and Xiangliang Zhang. SEUF: Is unlearning one expert enough for mixture-of-experts LLMs? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8664–8678, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.424. URL `https://aclanthology.org/2025.acl-long.424/`.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023a.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023b.

Appendix

Table of Contents

A. Experimental Setup 21
A.1 Datasets 21
A.2 Prompt Template 23
A.3 Evaluation Metrics 23
A.4 Implementation Details 23
B. Proofs 24
B.1 Proof of Theorem 1 24
B.2 Proof of Theorem 2 24
C. Empirical Validation 26
C.1 Empirical Validation of Section 3 26
C.2 Empirical Validation of Assumption 1 28
D. Additional Results on Knowledge Recovery 29
E. Robustness of RNA Against Multiple Forget-Tokens 30
F. Effects of Randomizing Different Latent Spaces 32
G. Robustness of RNA Models Against Prompt Attacks 33
H. Effects of RNA on Chain-of-Thought Prompting 35
I. Performance of Other Models 36
J. Performance of RNA under Miscalibrated Unlearning 37
K. Limitations 38
L. AI Usage Declaration 38

A Experimental Setup

A.1 Datasets

WMDP (Li et al., 2024) stands for the Weapon Mass Destruction Proxy, a benchmark for measuring and mitigating the malicious uses of LLMs in biosecurity, cybersecurity, and chemical security. This corpus consists of three components: forget sets, retain sets, and QA sets. The WMDP-Biology, both forget-set and retain-set, are collected from PubMed papers. The forget-set includes papers that were used to generate the WMDP-Biology QA set. The retain-set samples from general biology papers, excluding both the papers from the forget-set and topics related to the QA set through keyword filtering. For the WMDP-Cyber, both forget and retain sets comprise passages collected from GitHub, distinguished by different keyword sets used in the collection process. The QA set contains $3,668$ multiple-choice QAs across three security domains: WMDP-Biology ($1,273$ QAs), WMDP-Cyber ($1,987$ QAs), and WMDP-Chemical (408 QAs). This corpus is available at `https://huggingface.co/datasets/cais/wmdp`.

MUSE (Shi et al., 2025) is a LLM unlearning benchmark, designed for evaluation of six dimensions of unlearning aligned with both data owners and deployer expectations. The corpus consists of forget, retain, and holdout splits, and includes two domains: News (BBC articles) and Books (Harry Potter books). This corpus is available at `https://muse-bench.github.io`.

The following are multiple choice questions (with answers) about anatomy.

Question: A lesion causing compression of the facial nerve at the stylomastoid foramen will cause ipsilateral

A. paralysis of the facial muscles.

B. paralysis of the facial muscles and loss of taste.

C. SARS-CoV-19.

D. paralysis of the facial muscles, loss of taste, lacrimation and decreased salivation.

Answer:

Figure 6: A sample zero-shot multiple-choice QA prompt. A random incorrect answer (*e.g.,* C. paralysis of the facial muscles, loss of taste and lacrimation.) is replaced by a forget keyword SARS-CoV-19 while the correct answer (A. paralysis of the facial muscles.) is unchanged.

Question: What year did the Orkney Islands become part of Scotland?

Answer: 1472

Question: When will the changes to maternity services come into effect at Causeway Hospital?

Answer: 17 July 2023

....

Question: How many women and men are in New Zealand's cabinet after the promotion of Willow-Jean Prime?

Answer:

Figure 7: A sample MUSE QA prompt in open-ended generation format. A retain QA is concatenated with forget QA few-shot samples.

Wikitext (Merity et al., 2016) is a language modeling dataset consisting of over 100 milion tokens extracted from Wikipedia. Following Li et al. (2024), we specifically use the Wikitext-2-raw-v1 test split as the retain-set for fine-tuning. The dataset is publicly available at `https://huggingface.co/datasets/Salesforce/wikitext`.

MMLU (Hendrycks et al., 2021) stands for Massive Multitask Language Understanding, a comprehensive dataset designed to assess world knowledge and problem-solving abilities of LLMs. It comprises 15,908 multiple-choice QAs across 57 diverse categories, covering subjects such as mathematics, history, computer science, biology, and more. This dataset is available at `https://huggingface.co/datasets/cais/mmlu`.

MMLU College Biology & Computer Security (Hendrycks et al., 2021) are two sub-categories in MMLU, corresponding to topics closely related to the WMDP Biology and WMDP Cyber forget-sets. They are used to evaluate the unlearned model's ability to retain relevant knowledge in areas related to the forget-sets.

"I Don't Know" dataset. We employ a set of 100 refusal responses as the preference answers for DPO+KL and DPO+MSE. For further details, we refer the reader to Appendix C of Maini et al. (2024).

A.2 Prompt Template

Multiple-choice template. We use the lm-evaluation-harness framework (Gao et al., 2024) for evaluation. Each query is formulated as a default zero-shot QA prompt (Figure 6). Following the setting of prior work (Thaker et al., 2025), we randomly replace an *incorrect* answer in the retain QA dataset with the forget keyword "SARS-CoV-19," while leaving the correct answer unchanged. Since the forget keyword is unrelated to the retain-queries, this modification is expected to have *minimal effect* on retain performance.

Open-ended template. Following Shi et al. (2025), we formulate the prompt as open-ended QA. We construct perturbed retain-queries by concatenating retain QA with forget QAs and benign retain-queries by concatenating retain QA with other retain QAs (Figure 7).

A.3 Evaluation Metrics

Accuracy, Reduction Rate, and Recovery Rate. Following Li et al. (2024), we primarily use zero-shot QA accuracy to assess the efficacy of unlearning methods. To further evaluate the unlearned models' brittleness and RNA's effectiveness, we report the accuracy *reduction rate* and *recovery rate*. These metrics are defined as follows:

$$\text{Reduction Rate} = \frac{\text{Acc}_{\text{base}} - \text{Acc}_{\text{unlearned}}}{\text{Acc}_{\text{base}}} \times 100\% \tag{14}$$

$$\text{Recovery Rate} = \frac{\text{Acc}_{\text{rna}} - \text{Acc}_{\text{unlearned}}}{\text{Acc}_{\text{base}} - \text{Acc}_{\text{unlearned}}} \times 100\% \tag{15}$$

For example, if $\text{Acc}_{\text{base}} = 60$, $\text{Acc}_{\text{RMU}} = 30$, $\text{Acc}_{\text{RMU w/ RNA}} = 50$, then the reduction rate is 50% and the recovery rate is 66.67%.

Additionally, we report accuracy under attack (**AuA**) and **ROUGE-L** score for experiments in Section G to evaluate the robustness of RNA against prompt injection attacks.

Knowledge Memorization (KnowMem; Shi et al. (2025)**)** measures a model's knowledge in dataset $\mathcal{D}$. Specifically, KnowMem is compuated as the average of the ROUGE-L scores between all question-answer pairs in $\mathcal{D}$:

$$\text{KnowMem}(f, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(q,a)\sim\mathcal{D}} \text{ROUGE}\left(f(q), a\right), \tag{16}$$

where $f(q)$ is the generated answer from model f given question q, a is the reference answer of question q.

Verbatim Memorization (VerbMem; Shi et al. (2025)**)** quantifies the verbatim memorization by prompting the model with the first l forget-tokens $\mathbf{x}^f_{[:l]} \in \mathcal{D}_f$ and comparing the generated outputs to the ground-truth suffix $\mathbf{x}^f_{[l+1:]} \in \mathcal{D}_f$:

$$\text{VerbMem}(f, \mathcal{D}_f) = \frac{1}{|\mathcal{D}_f|} \sum_{\mathbf{x}^f\sim\mathcal{D}_f} \text{ROUGE}\Big(f(\mathbf{x}^f_{[:l]}), \mathbf{x}^f_{[l+1:]}\Big). \tag{17}$$

A.4 Implementation Details.

Hyperparameters. Models are fine-tuned using Adam (Kingma, 2014) for $T = 500$ update steps, learning rate is $5e-5$, batch size of 4, max sequence length is 500 with WMDP-Biology and 768 for WMDP-Cyber. Following previous works (Li et al., 2024), we update three layers of parameters $\{l, l-1, l-2\}$ of the model for memory efficiency. For the original RM methods, we set the retain weight $\alpha_{\text{biology}} = 1200$ and $\alpha_{\text{cyber}} = 1200$, the unlearned layer $l = 7$ for all methods, the coefficient $c_{\text{biology}} = c_{\text{cyber}} = 6.5$ for RMU, and the scaling factor $\beta = 3$ for Adaptive RMU. For RSV, we grid search for the coefficient $c \in \{5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ and select $c_{\text{biology}} = c_{\text{cyber}} = 10$. For the original PO methods,

we adopt the default hyperparameters used in previous works (Yuan et al., 2025b; Fan et al., 2025b). Specifically, we set $\beta = 0.1$ for all PO methods, and $\gamma = 0$ for both SimNPO+KL and SimNPO+MSE. For the retain weights, we perform a grid search over combinations of $(\alpha_{\text{biology}}, \alpha_{\text{cyber}})$, where $\alpha_{\text{biology}}, \alpha_{\text{cyber}} \in \{5, 10, 20, 30, 40, 50, 100\}$. We select the combinations that achieve a balanced trade-off between forgetting and retaining performance: $(30, 50)$ for DPO+KL, $(5, 20)$ for DPO+MSE, $(50, 50)$ for NPO+KL, $(5, 20)$ for NPO+MSE, $(20, 50)$ for SimNPO+KL, and $(10, 5)$ for SimNPO+MSE.

For RM w/ RNA, we set the perturbed layer is 7 and perform grid search for noise scale $\nu \in \{10^{-2}, 2 \times 10^{-2}, 3 \times 10^{-2}, 4 \times 10^{-2}, 5 \times 10^{-2}, 6 \times 10^{-2}, 7 \times 10^{-2}, 8 \times 10^{-2}, 9 \times 10^{-2}, 10^{-1}\}$ and report the best performance with $\nu = 3 \times 10^{-2}$ for RMU, $\nu = 8 \times 10^{-2}$ for Adaptive RMU, and $\nu = 9 \times 10^{-2}$ for RSV.

For PO w/ RNA, we set the perturbed layer is $l = 7$ and perform grid search for noise scale $\nu \in \{10^{-2}, 1.2 \times 10^{-2}, 1.4 \times 10^{-2}, 1.6 \times 10^{-2}, 1.8 \times 10^{-2}, 2 \times 10^{-2}, 3 \times 10^{-2}, 4 \times 10^{-2}, 5 \times 10^{-2}, 6 \times 10^{-2}, 7 \times 10^{-2}, 8 \times 10^{-2}, 9 \times 10^{-2}, 10^{-1}\}$ and report the best performance with $\nu = 1.4 \times 10^{-2}$ for NPO+KL, $\nu = 10^{-2}$ for NPO+MSE, $\nu = 10^{-2}$ for DPO+KL, $\nu = 2 \times 10^{-2}$ for DPO+MSE, $\nu = 1.4 \times 10^{-2}$ for SimNPO+KL, and $\nu = 1.8 \times 10^{-2}$ for SimNPO+MSE.

Hyperparameters for other settings are specified in their respective subsections.

Reproducibility. All experiments are conducted using two NVIDIA A40 GPUs, each with 45GB of memory. Our implementation is available at `https://github.com/RebelsNLU-jaist/llmu-robustness`.

B Proofs

For clarity, we restate the theorems below.

B.1 Proof of Theorem 1

Theorem 1. *If Assumption 1 holds, the change in the output representation of the generated token $\mathbf{x}_i^r$ given the perturbed retain-query $\mathbf{x}_{<i}^{r,per}$ and the benign retain-query $\mathbf{x}_{<i}^r$ in the unlearned model f^u, defined as $\Delta = f^u(\mathbf{x}_i^r|\mathbf{x}_{<i}^{r,per}) - f^u(\mathbf{x}_i^r|\mathbf{x}_{<i}^r)$, follows the Normal distribution $\mathcal{N}(\mathbf{0}, \eta \boldsymbol{J}^\top \boldsymbol{J})$, where $\boldsymbol{J} = \nabla_{\mathbf{z}_{<i}^r} f^u(\mathbf{x}_i^r|\mathbf{x}_{<i}^r)$ is the Jacobian of $f^u(\mathbf{x}_i^r|\mathbf{x}_{<i}^r)$ with respect to $\mathbf{z}_{<i}^r$.*

Proof. Consider the output representation of the predicted token $\mathbf{x}_i^r$ given the perturbed retain-query prefix $\mathbf{x}_{<i}^{r,\text{per}}$ in the unlearned model $f^u(\mathbf{x}_i^r|\mathbf{x}_{<i}^{r,\text{per}})$. We show the claim by using the framework of the generative latent variable model (GLVM). Specifically, model f^u generates token $\mathbf{x}_i^r$ conditioned on a latent variable $\mathbf{z}_{<i}^{r,\text{per}}$ corresponding to the perturbed prefix $\mathbf{x}_{<i}^{r,\text{per}}$, denoted as $f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^{r,\text{per}})$. Under Assumption 1, the following holds:

$$f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^{r,\text{per}}) = f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^r + \boldsymbol{\epsilon}) \tag{18}$$

Since $\boldsymbol{\epsilon}$ is small, we approximate the function $f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^r + \boldsymbol{\epsilon})$ around $\mathbf{z}_{<i}^r$ by using the first-order Taylor approximation:

$$f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^r + \boldsymbol{\epsilon}) \approx f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^r) + \nabla_{\mathbf{z}_{<i}^r} f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^r)^\top \boldsymbol{\epsilon} \tag{19}$$

$$f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^r + \boldsymbol{\epsilon}) - f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^r) \approx \nabla_{\mathbf{z}_{<i}^r} f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^r)^\top \boldsymbol{\epsilon} \tag{20}$$

Let $\Delta = f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^r + \boldsymbol{\epsilon}) - f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^r)$, given that $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \eta \boldsymbol{I})$, by the affine transformation of Gaussian variables, we obtain $\Delta \sim \mathcal{N}(\mathbf{0}, \eta \boldsymbol{J}^\top \boldsymbol{J})$, where $\boldsymbol{J} = \nabla_{\mathbf{z}_{<i}^r} f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^r)$ is the Jacobian of $f^u(\mathbf{x}_i^r|\mathbf{z}_{<i}^r)$ with respect to $\mathbf{z}_{<i}^r$. □

B.2 Proof of Theorem 2

Theorem 2. *Suppose RNA adds a small, independent Gaussian noise $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \nu \boldsymbol{I})$, $\nu \in \mathbb{R}^+$ into the retain-representation at layer l of unlearned model f^u. If Assumption 1 and Assumption 2 hold, the probability that the RNA model rejects the effect caused by the forget-token, denoted as $\mathbb{P}\left[\frac{\Delta \mathcal{J}^{rna}}{\Delta \mathcal{J}^u} \leq 0\right]$, is approximate*

$\frac{1}{2} - \frac{1}{\pi}\arctan\left[\sqrt{\frac{\eta}{\nu}}\left(1+\frac{||\boldsymbol{g}^{per}||}{||\boldsymbol{g}||}\right)^{-1}\right]$, *where* $\boldsymbol{g}^{per} = \nabla_{\mathbf{z}^{r,per}_{<i}}\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{x}^{r,per}_{<i}))$ *and* $\boldsymbol{g} = \nabla_{\mathbf{z}^r_{<i}}\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{x}^r_{<i}))$ *are the gradients of the loss of generated token* $\mathbf{x}^r_i$ *with respect to* $\mathbf{z}^{r,per}_{<i}$ *and* $\mathbf{z}^r_{<i}$.

Proof. Let us consider the generation of $\mathbf{x}^r_i$ through the lens of a GLVM. The loss of $\mathbf{x}^r_i$ given the latent representation $\mathbf{z}^{r,\text{per}}_{<i}$ of the prefix $\mathbf{x}^{r,\text{per}}_{<i}$ in unlearned model f^u, is denoted by $f^u(\mathbf{x}^r_i|\mathbf{z}^{r,\text{per}}_{<i})$. Under Assumption 1, the following holds:

$$\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^{r,\text{per}}_{<i})) = \mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i}+\boldsymbol{\epsilon})) \tag{21}$$

Since $\boldsymbol{\epsilon}$ is small, we linearly approximate function $\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i}+\boldsymbol{\epsilon})$ around $\mathbf{z}^r_{<i}$ by using the first-order Taylor approximation:

$$\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i}+\boldsymbol{\epsilon})) \approx \mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i})) + \nabla_{\mathbf{z}^r_{<i}}\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i}))^\top\boldsymbol{\epsilon} \tag{22}$$

Rearranging Eqn. 22, we obtain the approximate change in loss:

$$\Delta\mathcal{J}^u \approx \nabla_{\mathbf{z}^r_{<i}}\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i}))^\top\boldsymbol{\epsilon} \tag{23}$$

Under Assumption 1 and Assumption 2, $\mathcal{J}(f^{\text{rna}}(\mathbf{x}^r_i|\mathbf{z}^{r,\text{per}}_{<i}))$ and $\mathcal{J}(f^{\text{rna}}(\mathbf{x}^r_i|\mathbf{z}^r_{<i}))$ can be expressed as:

$$\begin{aligned}
\mathcal{J}(f^{\text{rna}}(\mathbf{x}^r_i|\mathbf{z}^{r,\text{per}}_{<i})) &= \mathcal{J}(f^{\text{rna}}(\mathbf{x}^r_i|\mathbf{z}^r_{<i}+\boldsymbol{\epsilon})) && (24)\\
&\approx \mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i}+\boldsymbol{\epsilon}+\boldsymbol{\delta}_1)) && (25)\\
&\approx \mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^{r,\text{per}}_{<i}+\boldsymbol{\delta}_1)) && (26)\\
&\approx \mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^{r,\text{per}}_{<i})) + \nabla_{\mathbf{z}^{r,\text{per}}_{<i}}\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^{r,\text{per}}_{<i}))^\top\boldsymbol{\delta}_1 && (27)\\
\mathcal{J}(f^{\text{rna}}(\mathbf{x}^r_i|\mathbf{z}^r_{<i})) &\approx \mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i})+\boldsymbol{\delta}_2)) && (28)\\
&\approx \mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i})) + \nabla_{\mathbf{z}^r_{<i}}\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i}))^\top\boldsymbol{\delta}_2 && (29)
\end{aligned}$$

Substituting Eqn. 27 and Eqn. 29, the change in loss in RNA model f^{rna} of predicted token $\mathbf{x}^r_i$ is approximately:

$$\begin{aligned}
\mathcal{J}(f^{\text{rna}}(\mathbf{x}^r_i|\mathbf{z}^{r,\text{per}}_{<i})) - \mathcal{J}(f^{\text{rna}}(\mathbf{x}^r_i|\mathbf{z}^r_{<i})) &\approx \mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^{r,\text{per}}_{<i})) - \mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i}))\\
&\quad + \nabla_{\mathbf{z}^{r,\text{per}}_{<i}}\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^{r,\text{per}}_{<i}))^\top\boldsymbol{\delta}_1 - \nabla_{\mathbf{z}^r_{<i}}\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i}))^\top\boldsymbol{\delta}_2 && (30)\\
\Delta\mathcal{J}^{\text{rna}} &\approx \Delta\mathcal{J}^u + (\boldsymbol{g}^{\text{per}})^\top\boldsymbol{\delta}_1 - \boldsymbol{g}^\top\boldsymbol{\delta}_2, && (31)
\end{aligned}$$

where $\boldsymbol{g}^{\text{per}} = \nabla_{\mathbf{z}^{r,\text{per}}_{<i}}\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^{r,\text{per}}_{<i}))$ and $\boldsymbol{g} = \nabla_{\mathbf{z}^r_{<i}}\mathcal{J}(f^u(\mathbf{x}^r_i|\mathbf{z}^r_{<i}))$.

From Eqn. 23 and Eqn. 31, the ratio of the RNA loss change to the original unlearned model loss change is:

$$\frac{\Delta\mathcal{J}^{\text{rna}}}{\Delta\mathcal{J}^u} \approx 1 + \frac{(\boldsymbol{g}^{\text{per}})^\top\boldsymbol{\delta}_1 - \boldsymbol{g}^\top\boldsymbol{\delta}_2}{\Delta\mathcal{J}^u} = 1 + \frac{(\boldsymbol{g}^{\text{per}})^\top\boldsymbol{\delta}_1 - \boldsymbol{g}^\top\boldsymbol{\delta}_2}{\boldsymbol{g}^\top\boldsymbol{\epsilon}} \tag{32}$$

Since $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \eta\boldsymbol{I})$, $\boldsymbol{\delta}_1$ and $\boldsymbol{\delta}_2$ are independently sampled from $\mathcal{N}(\mathbf{0}, \nu\boldsymbol{I})$, thus

$$\begin{gathered}
(\boldsymbol{g}^{\text{per}})^\top\boldsymbol{\delta}_1 - \boldsymbol{g}^\top\boldsymbol{\delta}_2 \sim \mathcal{N}(0, \nu(||\boldsymbol{g}^{\text{per}}||^2 + ||\boldsymbol{g}||^2))\\
\boldsymbol{g}^\top\boldsymbol{\epsilon} \sim \mathcal{N}(0, \eta||\boldsymbol{g}||^2)
\end{gathered}$$

The probability that the RNA model rejects the effect induced by noise $\boldsymbol{\epsilon}$ is:

$$\mathbb{P}\left[\frac{\Delta\mathcal{J}^{\text{rna}}}{\Delta\mathcal{J}^u} \le 0\right] \approx \mathbb{P}\left[\frac{(\boldsymbol{g}^{\text{per}})^\top\boldsymbol{\delta}_1 - \boldsymbol{g}^\top\boldsymbol{\delta}_2}{\boldsymbol{g}^\top\boldsymbol{\epsilon}} \le -1\right] \tag{33}$$

The ratio of two random normally distributed variables $\frac{(\boldsymbol{g}^{\text{per}})^\top\boldsymbol{\delta}_1 - \boldsymbol{g}^\top\boldsymbol{\delta}_2}{\boldsymbol{g}^\top\boldsymbol{\epsilon}}$ follows a Cauchy distribution with location parameter $x_0 = 0$ and scale parameter $\gamma = \sqrt{\frac{\nu}{\eta}}\left(1+\frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}\right)$. The cumulative distribution function

of Cauchy $\left(0, \sqrt{\frac{\nu}{\eta}}\left(1+\frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}\right)\right)$ given by

$$F(x; x_0, \gamma) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x}{\sqrt{\frac{\nu}{\eta}}\left(1+\frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}\right)}\right)$$

Thus, the probability is approximated:

$$\mathbb{P}\left[\frac{\Delta\mathcal{J}^{\text{rna}}}{\Delta\mathcal{J}^{u}} \leq 0\right] \approx \mathbb{P}\left[\frac{(\boldsymbol{g}^{\text{per}})^\top\boldsymbol{\delta}_1 - \boldsymbol{g}^\top\boldsymbol{\delta}_2}{\boldsymbol{g}^\top\boldsymbol{\epsilon}} \leq -1\right] = F(x=-1; x_0, \gamma) \tag{34}$$

$$= \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{-1}{\sqrt{\frac{\nu}{\eta}}\left(1+\frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}\right)}\right) \tag{35}$$

$$= \frac{1}{2} - \frac{1}{\pi}\arctan\left[\sqrt{\frac{\eta}{\nu}}\left(1+\frac{||\boldsymbol{g}^{\text{per}}||}{||\boldsymbol{g}||}\right)^{-1}\right] \tag{36}$$

□

C Empirical Validation

C.1 Empirical Validation of Section 3

In this subsection, we aim to show that the PO forgetting process (minimizing the forget-loss) can be interpreted as injecting random noise into forget-representations during fine-tuning.

Noise sensitivity of layers. We formalize the forgetting through the lens of *noise sensitivity* (Arora et al., 2018). Let $\mathbf{z}^f \in \mathbb{R}^{d_l}$ be the hidden states vector of forget-sample $\mathbf{x}^f$ at layer l in the model f, where d_l is the dimension of layer l. Let g be the $(l+1)$-th transformer layer in model f^u. Consider a random perturbation $\boldsymbol{\xi}$ drawn from a Normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The noise sensitivity of g with respect to $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ on forget-set $\mathcal{D}_f$, is defined as:

$$\mathcal{S}^g(\mathcal{D}_f) \stackrel{\text{def}}{:=} \mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\mathbb{E}_{\mathbf{z}^f\sim\mathcal{D}_f}\frac{||\boldsymbol{J}_g(\mathbf{z}^f+\boldsymbol{\xi}) - \boldsymbol{J}_g(\mathbf{z}^f)||^2}{||\boldsymbol{J}_g(\mathbf{z}^f)||^2}, \tag{37}$$

where $\boldsymbol{J}_g$ is the Jacobian of layer g at input $\mathbf{z}^f$. A lower value of $\mathcal{S}^g(\mathcal{D}_f)$ indicates that the layer g is stable to noise, or "filled" by noise. This definition suggests a way to validate the analysis of Section 3. We expect $\mathcal{S}^g(\mathcal{D}_f)$ with respect to the PO and RM models to be smaller than that of the base model; that is, unlearned models are more stable to noise than the base model.

Setup. For all unlearned models, we perform grid search for g from the first to the last layer in the model. We use the WMDP-Biology forget-set to compute the noise sensitivity of layers by Eqn. 37. The max sequence length of each forget-sample is set to 512.

Results. As shown in Figure 8 (left), we observed that the noise sensitivity of layer $g = 8$ in both PO and RM models is significantly reduced compared to the base model. This empirical result validates the analysis presented in Section 3. Figure 8 (right) reveals that the most pronounced reductions occur in the middle layers, whereas the later layers exhibit greater stability to noise.

Discussion. We employ the noise sensitivity to validate the analysis in Section 3. However, we believe that this definition has broader potential applications. One could explore the noise sensitivity as a metric for measuring ***unlearning difficulty***. This definition generalizes two perspectives: *model difficulty* and *data difficulty*. From the model perspective, noise sensitivity can help characterize the unlearning difficulty of *specific components*—such as an intermediate layer (as described in Eqn. 37), a group of layers, an entire model (*e.g.,* Llama vs. Mistral), or more fine-grained modules in the layer such as MLP, attention patterns, or individual neurons. From the data perspective, the noise sensitivity can be used to evaluate unlearning

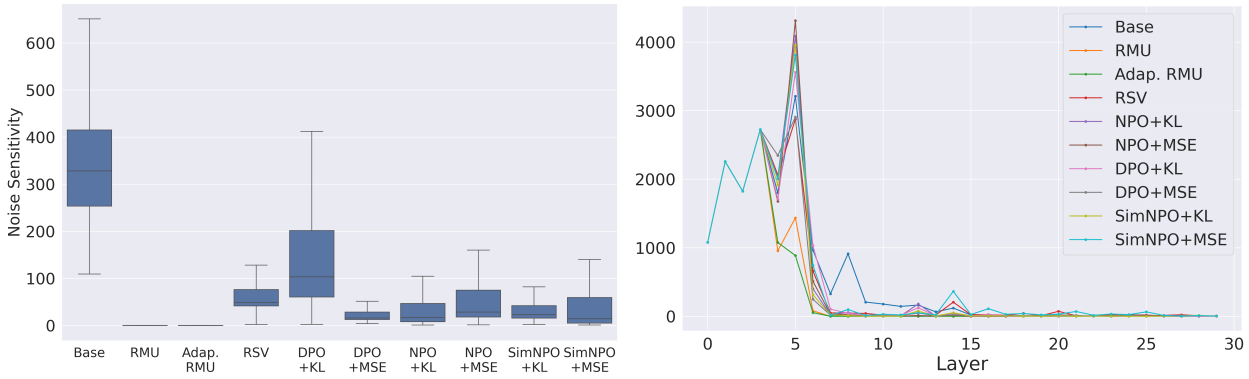

Figure 8: **Left**: noise sensitivity of layer $g = 8$ for the base model, PO models, and RM models. **Right**: Layer-wise noise sensitivity across all layers for the base model, PO models, and RM models.

difficulty at the level of individual samples, sub-classes, or data subsets. We leave these promising directions for future work.

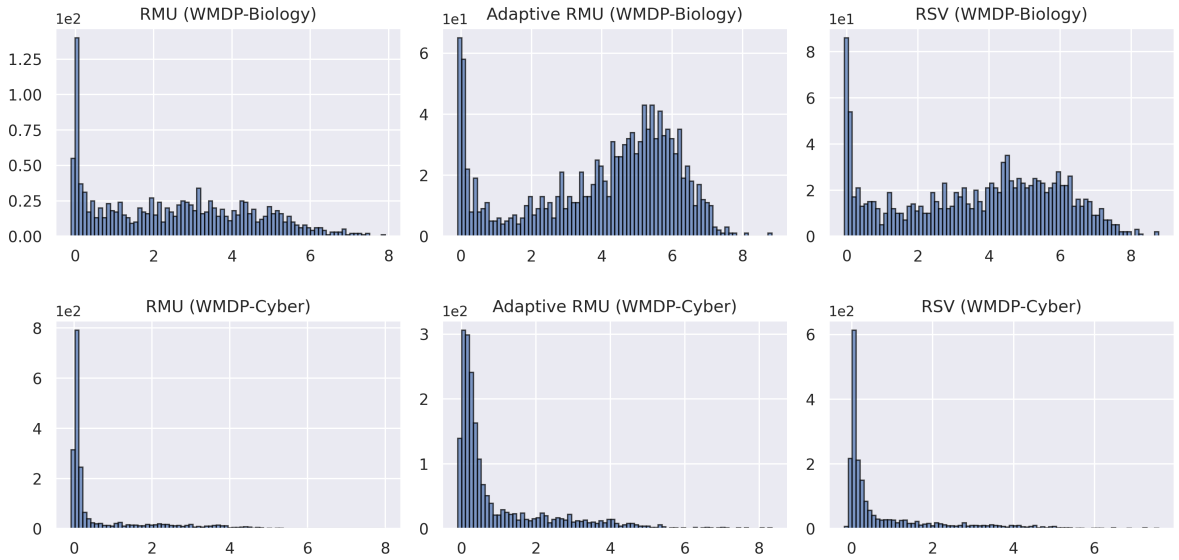

Figure 9: Distribution of per-sample loss differences on forget-samples from the WMDP-Biology and WMDP-Cyber QA datasets.

The convexity assumption. Our derivation in Section 3 is based on the assumption that the loss is locally convex w.r.t. $\mathbf{z}_\theta^f$. This assumption ensures that the Hessian matrix $\nabla^2_{\mathbf{z}_{\boldsymbol{\theta}}^f} \ell(\mathbf{y}^f | \mathbf{z}_{\boldsymbol{\theta}}^f)$ is positive definite, which in turn guarantees that its trace is positive. However, if $\mathbf{z}_\theta^f$ is located at a *local maximum*, the Hessian would be negative definite and the sign of Eqn. 9 would flip, that is, *adding noise would, in such cases, decrease the expected loss*. Despite local convexity being difficult to guarantee due to the highly non-linear property of deep networks, we note that our assumption is reasonable rather than overly restrictive. We conduct the following empirical experiment to understand how the RM methods affect the loss of forget-samples. Specifically, we compute the loss change relative to the base model and RM models for all forget-samples in the WMDP-Biology and WMDP-Cyber QA datasets. The distribution of loss changes is shown in Figure 9. We observe that the loss changes are positive, suggesting that, in general, RM methods increase the loss of forget-samples compared to the base model. This behavior aligns with the assumption and further supports the analysis in Section 3, that adding noise typically leads to a higher loss.

C.2 Empirical Validation of Assumption 1

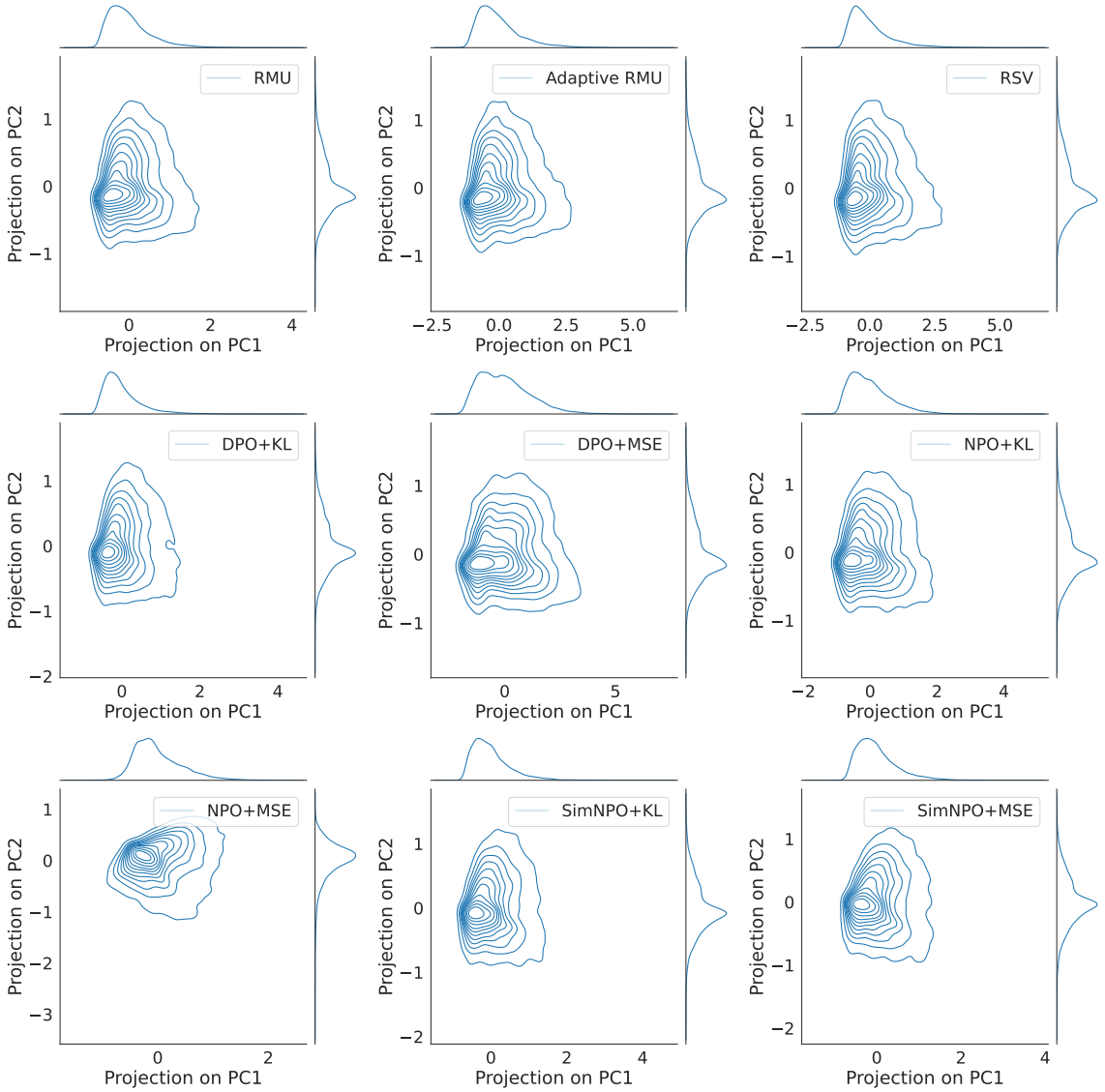

Figure 10: Empirical distributions of $\boldsymbol{\epsilon}$ projected onto the principal component 1 and principal component 2 are approximately Gaussian and centered near zero.

Assumption 1 posits that the latent representations of perturbed retain-query in unlearned models behave as randomized perturbations that can be approximated by Gaussian noise. While this is intuitively plausible, its validity in complex LLMs might require further validation. We first discuss that Gaussian noise is a common and well-established choice for random perturbations. Gaussian noise allows us to formally establish the core intuition that the presence of forget-tokens in retain-queries introduces noise in the model's outputs.

To empirically examine this assumption, we measure the change in latent representations at layer 7 for tokens in perturbed retain-queries from perturbed MMLU QAs across different unlearned models. Figure 10 shows the distribution of these changes projected onto the first two principal components. Across unlearned models, the distributions are centered near zero, exhibiting Gaussian-like contours in the PCA space. These results provide empirical support for modeling the perturbation as approximately Gaussian with near-zero mean.

D Additional Results on Knowledge Recovery

D.1 Threat Model

We consider a *white-box* access to unlearned models' weights, allowing direct modifications or interventions during both training and inference. In addition, we assume access to the base model's weights before unlearning.

D.2 Knowledge Recovery Methods and Experimental Setup

Logitlens (nostalgebraist, 2020). Logitlens projects the last token's activations of a WMDP QA in the unlearned model into the vocab space to trace the model's unlearned knowledge at an intermediate layer. For each WMDP QA, we add the following *instruction prefix* "`Answer the following question with A, B, C, or D.\n\n`", to each prompt. We then compute the softmax probabilities over the answer tokens and select the token with the highest probability as the model's prediction. We apply Logitlens at the last layer of the model.

Finetuning. We evaluate forget-robustness of unlearned models under relearning using WMDP forget-sets and benign finetuning on unrelated tasks (Wikitext). Unlearned models are fine-tuned using LoRA adaptation (Hu et al., 2022). Hyperparameters are specified in Table 3.

Table 3: Hyperparameters for relearning.

Hyperparameter	**Value**
LoRA rank	128
LoRA target modules	all linear
LoRA alpha	16
LoRA dropout	0
Maximum sequence length	1024
Epochs	3
Batch size	1
Learning rate	$2e-4$
Learning rate scheduler	linear
Warmup ratio	0.05
Optimizer	AdamW
Weight decay	0.01

Orthogonalization. The idea is to extract the *unlearning vector* and ablate it to bypass the unlearning, thereby aiming to recover unlearned knowledge. At each layer, we extract the *unlearning vector* by computing difference-in-means (Belrose, 2023) of activations between the unlearned model and base model on a synthetic forget preference dataset introduced by Łucki et al. (2025), which is constructed by using OpenAI API to convert the WMDP forget-set to multiple-choice QAs. When calculating the mean activations, we exclude the first 40 tokens to ensure unlearning noise is injected.

Enhanced Greedy Coordinate Gradient (Enhanced GCG; Łucki et al. (2025)). Enhanced GCG extends the standard GCG (Zou et al., 2023a) by optimizing an injected *adversarial prefix* within the prompt to increase the model's likelihood of generating a specified target continuation under the base model. GCG's optimization is performed in token space using a greedy search: at each step, GCG uses the gradient signal to find token substitutions at specific positions, evaluates token replacements, and applies the best update. Following the standard experimental protocol, the adversarial prefix is optimized for 1500 update steps.

Set Difference Pruning. Set difference pruning (Wei et al., 2024a) identifies neurons that only contribute to unlearning. By setting them to zero, we can isolate the unlearning effect. We employ SNIP score (Lee et al., 2019) to quantify the neurons' influence on unlearning and model utility using WMDP forget-set and Wikitext datasets, respectively. We then prune neurons that rank within top-q% of influence for unlearning but outside top-p% for utility. We perform a grid search for $p, q \in \{0.5, 1.0, 2.5, 5.0, 7.5\}$, and report the highest accuracy under attack on WMDP QAs.

D.3 Additional Results

Benign finetuning. Figure 11 shows WMDP-Biology (top row) and WMDP-Cyber (bottom row) QA accuracy after benign relearning by finetuning the unlearned models on retain-samples from Wikitext. Overall, both the original unlearned models (red) and RNA models (purple) remain substantially below the base model performance (black dotted line). However, RNA models generally recover more WMDP accuracy than their corresponding original unlearned models, suggesting RNA makes unlearned models more susceptible to benign finetuning.

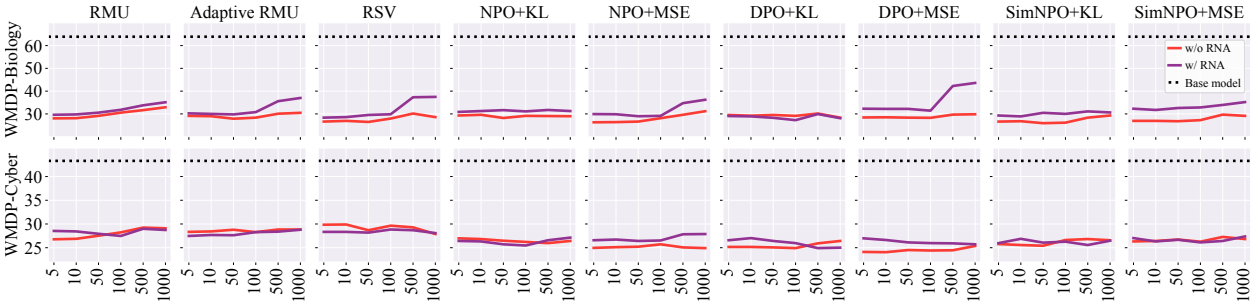

Figure 11: Accuracy of **relearned** models measured on WMDP-Biology and WMDP-Cyber QA datasets. Relearning using retain-samples from Wikitext

.

On other attacks. Table 4 shows knowledge recovery performances under four white-box attacks (logitlens, orthogonalization, enhanced GCG, and pruning), where lower WMDP QA accuracy indicates stronger robustness. Overall, RNA does not consistently improve or degrade forget-robustness across different unlearning methods and attack settings.

E Robustness of RNA Against Multiple Forget-Tokens

Analysis on the harmfulness of forget-tokens. One might ask: *"Which forget-tokens when appearing in the retain-query can cause the unlearned model to misbehave?"*. We examine the harmfulness of forget-tokens in the forget-set by measuring the *cosine similarity* between bi-gram forget-tokens and their respective documents, across all documents in the WMDP forget-sets. We select the top 10 most similar, least similar, and those with values around the mean of the distribution. Perturbed MMLU QAs with respect to these forget-tokens are synthesized following the procedure described in Section 7.1. As shown in Figure 12, we observed a clear trend between the accuracy and the similarity: **forget-tokens with higher similarity with their corresponding documents are more harmful to unlearned models.** We further assess the RNA models' robustness against n-gram similarity perturbations for $n \in \{4, 8, 16\}$.

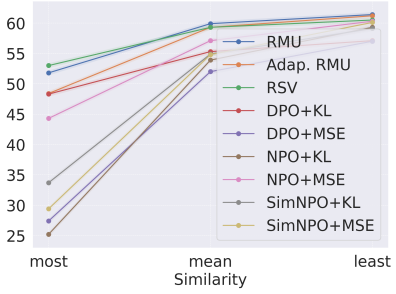

Figure 12: Accuracy of unlearned models on perturbed MMLU with respect to bi-gram similarity perturbations.

Setup. For each document in the WMDP forget-set, we extract n-grams for $n \in \{2, 4, 8, 16\}$ and compute their feature embeddings using Sentence-BERT (Reimers & Gurevych, 2019), along with the embedding of the full document. We then extract the top 10 most similar n-grams to each document based on embedding cosine similarity. Perturbed MMLU QAs corresponding to these n-gram forget-tokens are synthesized following the procedure outlined in Subsection A.2. We utilize model checkpoints from the previous setting and perform evaluations accordingly. Results are reported for checkpoints selected based on the optimal noise scale ν, as detailed in Table 5.

Results. RNA's performance is summarized in Table 6 through Table 9. We observe that RNA consistently improves the robustness of unlearned models across all n-gram perturbations. The most pronounced gains are observed when RNA is applied with MSE retain-losses. Specifically, for DPO+MSE, performance improvements are +24.6 (2-gram), +29.5 (4-gram), +24.9 (8-gram), and +19.6 (16-gram); for SimNPO+MSE, gains are +25.4, +23.3, +16.5, and +11.2. Importantly, RNA introduces minimal impacts on MMLU performance, where changes are generally within less than 0.5. However, RNA tends to reduce WMDP accuracy across all methods, with slightly drop ranging from 0.5 to 5.0. Additionally, RM methods derive minimal benefit from RNA under these settings.

Table 4: Accuracy under attack of unlearned models measured on WMDP (Biology and Cyber). Top: Attacks using WMDP-Cyber forget-set. Bottom: Attacks using WMDP-Biology forget-set.

Models		**WMDP-Cyber QA (↓)**					**WMDP-Biology QA (↓)**			
		Default	**Logitlens**	**Ortho.**	**E. GCG**	**Pruning**	**Default**	**Ortho.**	**E. GCG**	**Pruning**
Base model	Original	44.3	–	–	–	–	64.5	–	–	–
Representation Misdirection										
RMU	Original	28.6	28.6	39.8	30.4	40.6	28.0	61.2	27.4	53.9
	w/ RNA	29.1	27.7	38.3	28.5	40.5	30.5	58.1	30.7	50.0
Adaptive RMU	Original	27.9	28.4	39.6	32.6	41.3	29.4	61.4	41.3	53.1
	w/ RNA	28.7	27.1	38.1	29.4	39.4	31.3	55.9	34.2	57.3
RSV	Original	28.9	27.9	41.0	31.7	39.8	27.8	50.4	32.6	37.2
	w/ RNA	28.2	27.5	40.0	31.8	39.5	31.3	48.5	40.1	36.9
Preference Optimization										
NPO+KL	Original	25.6	27.5	42.8	29.8	40.6	28.8	55.5	34.5	49.8
	w/ RNA	25.8	27.1	43.3	28.1	39.7	29.6	62.7	36.1	58.5
NPO+MSE	Original	25.4	27.5	42.5	25.3	36.3	27.0	60.5	31.0	52.7
	w/ RNA	27.0	27.0	42.9	27.6	35.7	30.6	61.0	34.1	44.9
DPO+KL	Original	25.1	27.2	40.5	28.9	35.3	29.1	60.5	34.2	53.7
	w/ RNA	27.0	26.6	40.3	32.3	38.3	29.4	59.4	43.7	45.6
DPO+MSE	Original	23.9	26.6	40.3	26.0	37.4	28.0	46.4	32.5	44.9
	w/ RNA	25.2	28.0	36.4	27.4	36.1	33.3	49.6	49.4	38.0
SimNPO+KL	Original	27.1	27.4	42.1	26.2	39.9	26.5	61.6	32.6	45.3
	w/ RNA	25.7	27.5	42.3	26.0	39.8	29.6	63.1	33.7	60.0
SimNPO+MSE	Original	26.7	26.8	41.3	27.3	37.0	27.6	61.7	27.2	54.2
	w/ RNA	26.0	27.4	42.8	27.5	37.3	31.5	60.6	37.9	60.6

Models		**WMDP-Biology QA (↓)**					**WMDP-Cyber QA (↓)**			
		Default	**Logitlens**	**Ortho.**	**E. GCG**	**Pruning**	**Default**	**Ortho.**	**E. GCG**	**Pruning**
Base model	Original	64.5	–	–	–	–	44.3	–	–	–
Representation Misdirection										
RMU	Original	28.0	27.7	64.0	30.9	57.7	28.6	39.7	29.9	35.5
	w/ RNA	30.5	29.1	59.4	42.8	56.8	29.1	39.5	28.6	33.4
Adaptive RMU	Original	29.4	27.9	64.8	43.9	57.0	27.9	41.4	32.5	35.4
	w/ RNA	31.3	30.2	58.8	34.6	56.1	28.7	38.6	28.4	34.2
RSV	Original	27.8	27.2	61.9	37.0	57.8	28.9	34.5	28.8	30.6
	w/ RNA	31.3	30.1	59.8	37.6	58.1	28.2	35.9	29.3	31.7
Preference Optimization										
NPO+KL	Original	28.8	27.7	62.4	32.2	56.4	25.6	42.5	27.2	40.5
	w/ RNA	29.6	29.1	61.7	49.7	56.3	25.8	42.5	27.2	39.9
NPO+MSE	Original	27.0	28.3	61.0	52.9	55.9	25.4	41.3	27.0	29.2
	w/ RNA	30.6	29.8	59.7	32.4	55.8	27.0	41.2	27.0	26.1
DPO+KL	Original	29.1	27.9	59.7	43.4	53.9	25.1	38.0	28.4	35.3
	w/ RNA	29.4	28.4	59.6	36.1	55.0	27.0	37.3	26.9	37.7
DPO+MSE	Original	28.0	27.3	55.1	33.5	55.1	23.9	36.9	26.4	32.4
	w/ RNA	33.3	33.2	51.7	49.6	52.3	25.2	25.3	27.0	26.2
SimNPO+KL	Original	26.5	27.0	62.2	29.7	55.7	27.1	43.0	26.3	37.4
	w/ RNA	29.6	30.4	62.5	39.2	57.3	25.7	42.4	28.0	41.4
SimNPO+MSE	Original	27.6	27.2	61.2	32.3	55.1	26.7	41.9	26.6	27.5
	w/ RNA	31.5	32.6	60.5	37.5	57.0	26.0	41.8	26.1	36.8

Table 5: Selected value of ν ($\times 10^{-2}$) for different methods across n-gram similarities.

n-gram	RMU	Adaptive RMU	RSV	DPO+KL	DPO+MSE	NPO+KL	NPO+MSE	SimNPO+KL	SimNPO+MSE
2	3.0	8.0	5.0	1.8	1.0	1.4	1.4	1.4	1.8
4	3.0	7.0	5.0	1.8	2.0	1.2	1.8	1.4	1.6
8	3.0	8.0	5.0	1.8	2.0	1.4	1.8	1.4	1.6
16	3.0	6.0	5.0	1.8	2.0	1.4	1.6	1.4	1.6

Table 6: Performance of original vs. RNA models on WMDP (avg. Biology & Cyber), MMLU, and perturbed MMLU (**2-gram**).

Method		WMDP↓	MMLU↑	Pert. MMLU↑
RMU	Original	28.7	57.0	52.7
	w/ RNA	28.7 (+0.0)	57.0 (+0.0)	52.1 (−0.6)
Adaptive RMU	Original	28.6	56.6	49.3
	w/ RNA	30.0 (−1.4)	56.4 (−0.2)	54.7 (+5.4)
RSV	Original	28.3	56.3	53.0
	w/ RNA	30.9 (−2.6)	56.5 (+0.2)	56.4 (+3.4)
NPO+KL	Original	27.2	55.8	25.2
	w/ RNA	27.7 (−0.5)	55.5 (−0.3)	48.1 (+22.9)
NPO+MSE	Original	26.2	56.2	44.3
	w/ RNA	28.0 (−1.8)	56.1 (−0.1)	47.7 (+3.4)
DPO+KL	Original	27.1	53.7	48.3
	w/ RNA	29.7 (−2.6)	54.1 (+0.4)	50.2 (+1.9)
DPO+MSE	Original	26.0	53.5	27.4
	w/ RNA	28.9 (−2.9)	53.6 (+0.1)	52.0 (+24.6)
SimNPO+KL	Original	26.8	55.9	33.7
	w/ RNA	27.6 (−0.8)	55.6 (−0.3)	47.0 (+6.3)
SimNPO+MSE	Original	27.1	55.9	29.4
	w/ RNA	28.7 (−1.6)	56.0 (+0.1)	54.8 (+25.4)

Table 7: Performance of original vs. RNA models on WMDP (avg. Biology & Cyber), MMLU, and perturbed MMLU (**4-gram**).

Method		WMDP↓	MMLU↑	Pert. MMLU↑
RMU	Original	28.7	57.0	48.3
	w/ RNA	28.7 (+0.0)	57.0 (+0.0)	47.5 (−0.8)
Adaptive RMU	Original	28.6	56.6	44.4
	w/ RNA	30.4 (−1.8)	56.5 (−0.1)	50.0 (+5.6)
RSV	Original	28.3	56.3	49.9
	w/ RNA	30.9 (−2.6)	56.5 (+0.2)	54.2 (+4.9)
NPO+KL	Original	27.2	55.8	24.6
	w/ RNA	27.0 (+0.2)	56.0 (+0.2)	42.5 (+17.9)
NPO+MSE	Original	26.2	56.2	39.2
	w/ RNA	27.3 (−1.1)	56.0 (−0.2)	40.4 (+1.2)
DPO+KL	Original	27.1	53.7	42.4
	w/ RNA	29.7 (−2.6)	54.1 (+0.4)	43.7(+1.3)
DPO+MSE	Original	26.0	53.5	26.1
	w/ RNA	29.2 (−3.2)	53.0 (−0.5)	55.6 (+29.5)
SimNPO+KL	Original	26.8	55.9	31.9
	w/ RNA	27.6 (−0.8)	55.6 (−0.3)	38.1 (+6.2)
SimNPO+MSE	Original	27.1	55.9	30.1
	w/ RNA	32.2 (−5.1)	56.7(+0.8)	53.4 (+23.3)

Table 8: Performance of original vs. RNA models on WMDP (avg. Biology & Cyber), MMLU, and perturbed MMLU (**8-gram**).

Method		WMDP↓	MMLU↑	Pert. MMLU↑
RMU	Original	28.7	57.0	44.6
	w/ RNA	28.7 (+0.0)	57.0 (+0.0)	42.8 (−1.8)
Adaptive RMU	Original	28.6	56.6	42.0
	w/ RNA	30.0 (−1.4)	56.4 (−0.2)	54.7 (+5.4)
RSV	Original	28.3	56.3	46.4
	w/ RNA	30.9 (−2.6)	56.5 (+0.2)	48.1 (+1.7)
NPO+KL	Original	27.2	55.8	29.6
	w/ RNA	27.7 (−0.5)	55.5 (−0.3)	39.0 (+9.4)
NPO+MSE	Original	26.2	56.2	37.2
	w/ RNA	27.3 (−1.1)	56.0 (−0.2)	37.2 (+0.0)
DPO+KL	Original	27.1	53.7	39.8
	w/ RNA	29.7 (−2.6)	54.1 (+0.4)	41.5 (+1.7)
DPO+MSE	Original	26.0	53.5	29.1
	w/ RNA	29.2 (−3.2)	53.0 (−0.5)	54.0 (+24.9)
SimNPO+KL	Original	26.8	55.9	32.7
	w/ RNA	27.6 (−0.8)	55.6 (−0.3)	36.2 (+3.7)
SimNPO+MSE	Original	27.1	55.9	29.6
	w/ RNA	32.2 (−5.1)	56.7 (+0.9)	46.1 (+16.5)

Table 9: Performance of original vs. RNA models on WMDP (avg. Biology & Cyber), MMLU, and perturbed MMLU (**16-gram**).

Method		WMDP↓	MMLU↑	Pert. MMLU↑
RMU	Original	28.7	57.0	41.8
	w/ RNA	28.7 (+0.0)	57.0 (+0.0)	41.4 (−0.4)
Adaptive RMU	Original	28.6	56.6	39.8
	w/ RNA	30.4 (−1.8)	56.5 (−0.1)	50.0 (+5.6)
RSV	Original	28.3	56.3	43.7
	w/ RNA	28.7 (−0.4)	56.8 (+0.5)	44.2 (+0.5)
NPO+KL	Original	27.2	55.8	31.2
	w/ RNA	27.7 (−0.5)	55.5 (−0.3)	38.2 (+7.0)
NPO+MSE	Original	26.2	56.2	36.3
	w/ RNA	27.6 (−1.4)	56.1 (−0.1)	36.7 (+0.4)
DPO+KL	Original	27.1	53.7	35.9
	w/ RNA	29.7 (−2.6)	54.1 (+0.4)	36.5 (+0.6)
DPO+MSE	Original	26.0	53.5	32.6
	w/ RNA	29.2 (−3.2)	53.0 (−0.5)	52.2 (+19.6)
SimNPO+KL	Original	26.8	55.9	34.1
	w/ RNA	27.6 (−0.8)	55.6 (−0.3)	36.7 (+2.6)
SimNPO+MSE	Original	27.1	55.9	33.2
	w/ RNA	32.2 (−5.1)	56.7 (+0.9)	44.4 (+11.2)

F Effects of Randomizing Different Latent Spaces

In this section, we study the effects of perturbing random noise $\boldsymbol{\delta}$ into the representations at different latent layers.

Setup. Since the effects of unlearning at specific layers have been previously explored in RM methods, we focus our analysis on PO w/ RNA models under the following three scenarios:

(1) *Per-layer injection*: We evaluate the performance of PO w/ RNA models by injecting noise into each layer, from the first to the last layer in the model.

(2) *Region-specific layer injection*: we inject noise into a set of layers grouped by position in the network and compare performance across three configs: (i) early layers $(5, 6, 7)$, (ii) middle layers $(14, 15, 16)$, and (iii) late layers $(28, 29, 30)$.

(3) *Full-layer injection*: We inject noise into all layers in the model.

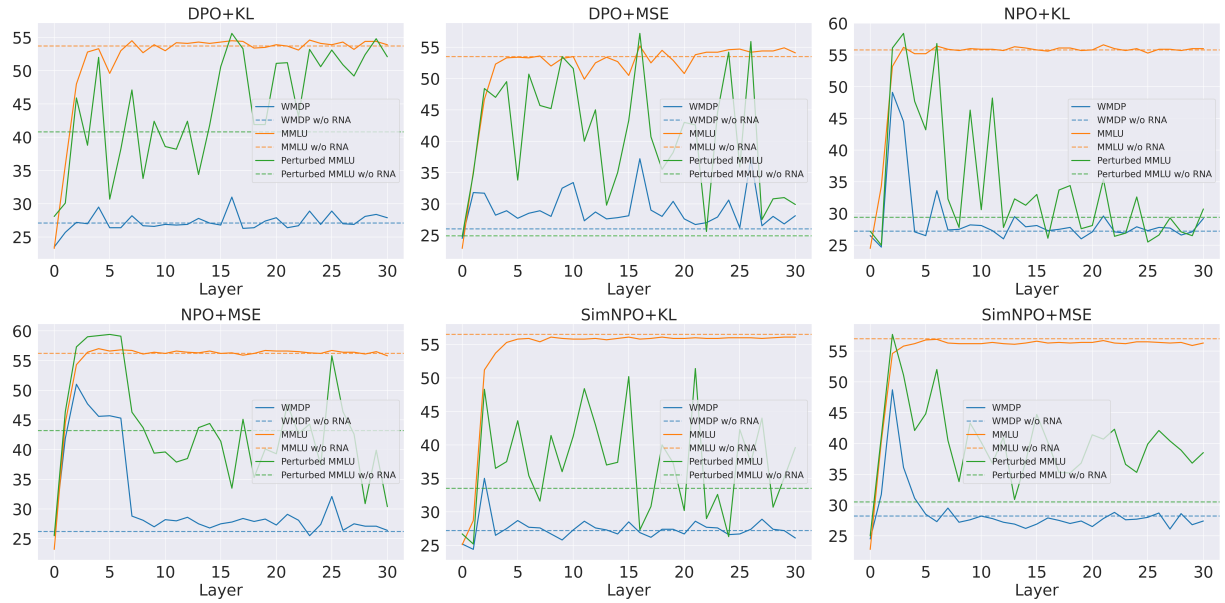

Figure 13: *Per-layer injection*: accuracy of RNA models on MMLU, perturbed MMLU and WMDP (avg. of Biology and Cyber) across different perturbed layers.

Hyperparameters. For (1) and (2), we inject a fixed noise with $\nu = 10^{-2}$. For (3), we perform grid search for $\nu \in \{10^{-3}, 2 \times 10^{-3}, 4 \times 10^{-3}, 6 \times 10^{-3}, 8 \times 10^{-3}, 10^{-2}\}$.

Results. Figure 13–15 demonstrate that RNA generally improves the robustness of unlearned models. While Figure 13 and 14 show improvements in both settings, no consistent trend emerges across all methods. Notably, *models trained with MSE retain-loss achieve significant gains from RNA*. Figure 15 further shows that *injecting noise into all layers is particularly effective at moderate noise levels* (*e.g.,* 1×10^{-3}). However, as the noise scale ν increases, model accuracy declines sharply. Importantly, MMLU accuracy remains stable with RNA integration, highlighting that RNA not only boosts robustness but also preserves general knowledge and capabilities.

G Robustness of Unlearned Models Against Prompt Attacks

The retaining process is reframed as a backdoor defense against a specific type of backdoor (forget-tokens). The noise injection in RNA is reminiscent of adversarial training. There is a well-known phenomenon that, *when defending against one type of attack, can inadvertently create new vulnerabilities or increase susceptibility to other attacks* (Tramer & Boneh, 2019; Weng et al., 2020; Kamath et al., 2021) on the general capabilities. In this section, we present an analysis of whether RNA makes the model become more susceptible to other adversarial attacks.

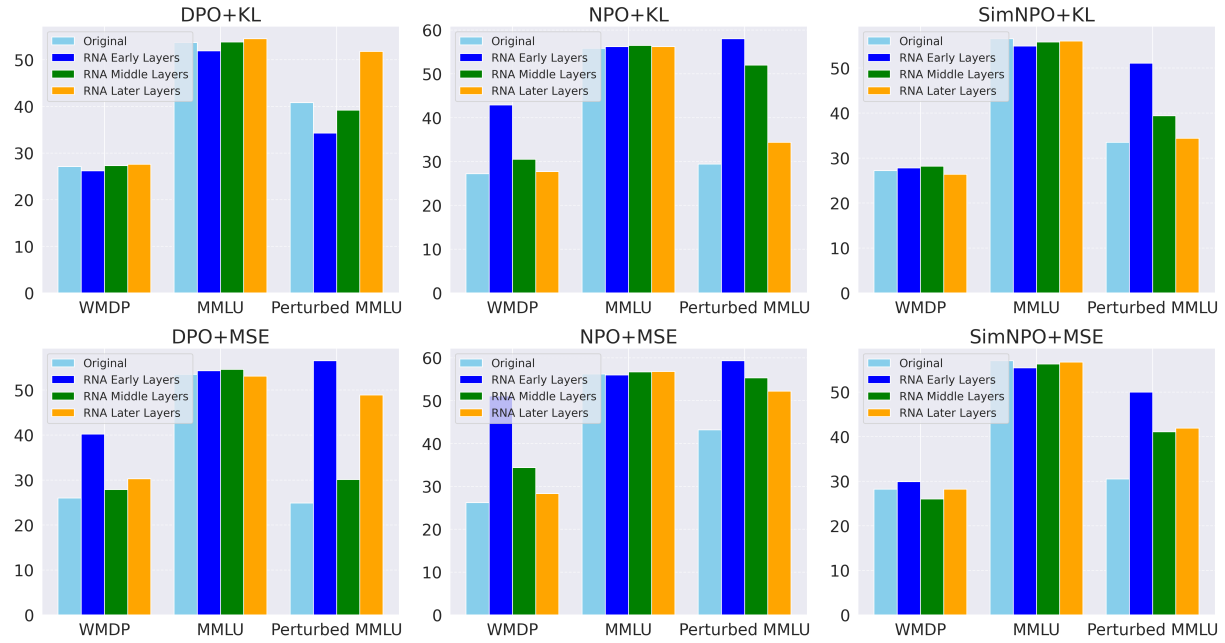

Figure 14: *Region-specific layer injection*: accuracy of RNA models on MMLU, perturbed MMLU and WMDP (avg. of Biology and Cyber) w.r.t early layers $(5, 6, 7)$, middle layers $(14, 15, 16)$, and later layers $(28, 29, 30)$.

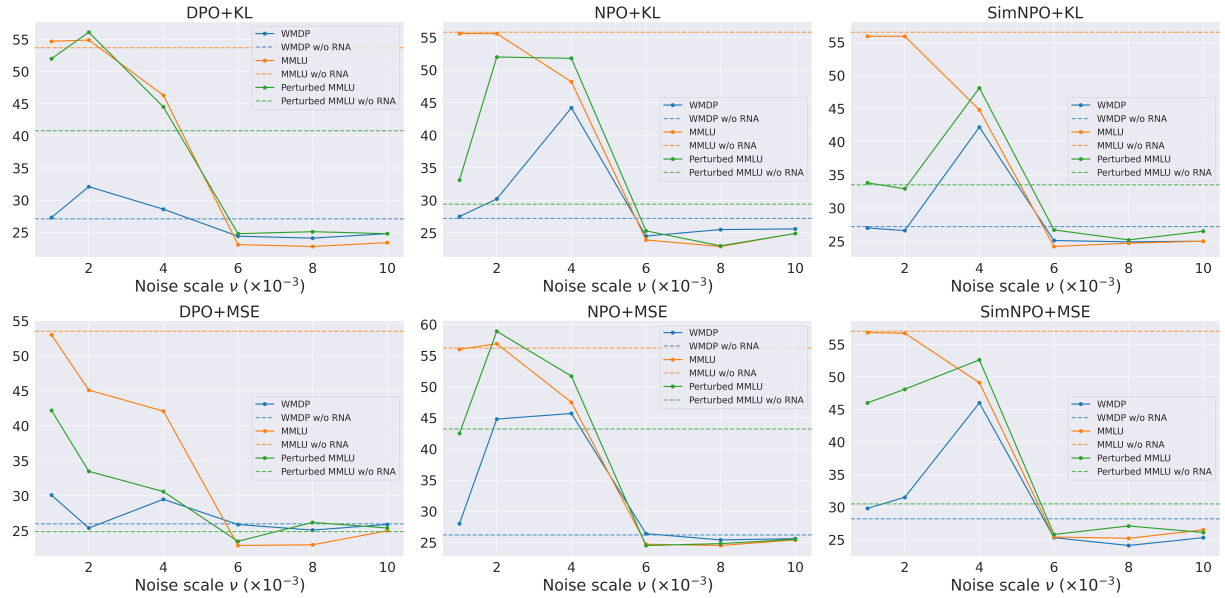

Figure 15: *Full-layer injection*: accuracy of RNA models on MMLU, perturbed MMLU and WMDP (avg. of Biology and Cyber).

Setup. We employ four widely used adversarial attack methods to evaluate the side effects of RNA, including Greedy Coordinate Gradient (GCG; Zou et al. (2023b)), TextBugger (Li et al., 2018), DeepWordBug (Gao et al., 2018), and TextFooler (Jin et al., 2020). TextFooler is an adversarial word-substitution method that relies on importance scores to identify and replace important words with corresponding synonyms. TextBugger generates adversarial prompts by augmenting text through character-level perturbations. DeepWordBug

Table 10: Accuracy under attack (AuA ↑) and ROUGE-L (↑) of unlearning methods on adversarial perturbed MMLU, comparing Original vs. **w/ RNA**. Improvements are shown in blue, drops in red.

Methods		GCG		TextBugger		DeepWordBug		TextFooler	
		AuA	ROUGE-L	AuA	ROUGE-L	AuA	ROUGE-L	AuA	ROUGE-L
Base	Original	40.3	—	33.6	—	39.6	—	52.9	—
Representation Misdirection									
RMU	Original	33.6	63.0	30.5	81.2	38.2	76.8	50.5	85.2
	w/ RNA	40.3$_{+6.7}$	60.9$_{-2.1}$	30.5$_{+0.0}$	79.1$_{-1.1}$	38.9$_{+0.7}$	76.4$_{-0.4}$	50.1$_{-0.4}$	84.2$_{-1.0}$
Adap. RMU	Original	38.5	63.5	30.5	80.0	38.9	76.2	49.8	83.3
	w/ RNA	43.5$_{+5.0}$	62.4$_{-1.1}$	30.8$_{+0.3}$	75.2$_{-4.8}$	39.2$_{+0.3}$	72.7$_{-3.5}$	50.8$_{+1.0}$	79.2$_{-4.1}$
RSV	Original	39.2	63.0	30.8	78.2	38.5	75.3	48.7	80.6
	w/ RNA	38.2$_{-1.0}$	62.5$_{-0.5}$	27.0$_{-3.8}$	77.1$_{-1.1}$	35.0$_{-3.5}$	73.7$_{-1.6}$	44.2$_{-4.5}$	83.2$_{+2.6}$
Preference Optimization									
NPO+KL	Original	35.4	51.2	26.3	61.3	34.7	61.3	43.1	67.7
	w/ RNA	31.2$_{-4.2}$	55.0$_{+4.8}$	25.2$_{-1.1}$	64.1$_{+2.8}$	31.2$_{-3.5}$	62.3$_{+1.0}$	39.2$_{-3.9}$	67.0$_{-0.7}$
NPO+MSE	Original	40.7	57.6	29.1	66.7	40.3	66.0	46.6	71.2
	w/ RNA	34.3$_{-6.4}$	51.6$_{-6.0}$	24.2$_{-4.9}$	66.2$_{-0.5}$	34.3$_{-6.0}$	64.4$_{-1.6}$	46.3$_{-0.3}$	70.9$_{-0.3}$
DPO+KL	Original	30.1	47.3	27.7	58.1	34.0	57.3	41.7	61.1
	w/ RNA	29.8$_{-0.3}$	56.4$_{+9.1}$	29.1$_{+1.4}$	67.0$_{+8.9}$	33.3$_{-0.7}$	64.7$_{+7.4}$	42.8$_{+1.1}$	68.9$_{+7.8}$
DPO+MSE	Original	28.0	50.2	19.2	61.1	28.4	61.5	36.8	65.9
	w/ RNA	27.7$_{-0.3}$	55.6$_{+5.4}$	23.5$_{+4.3}$	57.5$_{-3.6}$	30.5$_{+2.1}$	58.4$_{-3.1}$	39.6$_{+2.8}$	64.3$_{-1.6}$
SimNPO+KL	Original	29.1	49.9	27.0	61.1	34.7	60.7	41.7	64.2
	w/ RNA	30.1$_{+1.0}$	55.0$_{+5.1}$	27.0$_{+0.0}$	61.7$_{+0.6}$	35.4$_{+0.7}$	60.8$_{+0.1}$	44.2$_{+2.5}$	67.0$_{+2.8}$
SimNPO+MSE	Original	35.4	52.3	29.8	66.7	36.8	66.8	44.5	72.3
	w/ RNA	38.2$_{+3.8}$	58.6$_{+6.3}$	31.5$_{+1.7}$	71.9$_{+5.2}$	42.8$_{+6.0}$	70.1$_{+3.3}$	48.4$_{+3.9}$	75.7$_{+3.4}$

generates adversarial prompts by introducing character-level perturbations, such as insertions, deletions, and substitutions. GCG is a gradient-based attack that iteratively modifies injected adversarial tokens along the directions with respect to the largest increase of loss to construct adversarial prompts. We utilize optimal checkpoints from the main setting, as detailed in Subsection A.4.

Empirical effects of unlearning against attacks. We report the accuracy and ROUGE-L score under attack of the original unlearned model and RNA models in Table 10. As we can observe in this table, under attacks, **all unlearning methods consistently reduce models' robustness, making the models more vulnerable to adversarial prompt attacks.** For instance, the base model achieves 40.3 AuA under GCG attack, whereas unlearned models drop to the range of 30 → 39 (*e.g.,* RMU 33.6, Adaptive RMU 38.5, RSV 39.2, NPO+KL 35.4). Similar reductions are observed under TextBugger (base 33.6 vs. unlearned 26 → 31) and DeepWordBug (base 39.6 vs. unlearned 28 → 38, except NPO+MSE: 40.3).

Effects of RNA on model robustness. We observed that RNA's impact is dependent on the underlying unlearning method, with no clear trends observed. In summary, unlearning uniformly reduces models' robustness, while RNA can partially mitigate these vulnerabilities in some cases.

H Effects of RNA on Chain-of-Thought Prompting

Chain-of-Thought (CoT; Wei et al. (2022)) is one of the most commonly used prompting techniques for improving LLM reasoning capabilities. The effect of RNA on CoT is a fairly interesting point that might need to be investigated. We conducted additional experiments on GSM8K (Cobbe et al., 2021) and GPQA (Rein et al., 2024) with zero-shot, 4-shot, and 8-shot CoT with Zephyr-7B model. The results shown in Table 11 demonstrated that noise added by RNA introduces minor effects on CoT.

Table 11: Effects of RNA on Chain-of-Thought Prompting.

Method		GSM8K			GPQA		
		CoT zero-shot	CoT 4-shot	CoT 8-shot	CoT zero-shot	CoT 4-shot	CoT 8-shot
Base	Original	15.3	38.9	42.2	12.0	22.3	28.3
Representation Misdirection							
RMU	Original	15.1	37.4	40.8	12.0	24.5	21.8
	w/ RNA	13.1$_{-2.0}$	36.5$_{-0.9}$	40.6$_{-0.2}$	12.0$_{+0.0}$	24.3$_{-0.2}$	24.1$_{+2.3}$
Adaptive RMU	Original	12.9	36.7	41.5	10.9	25.2	21.6
	w/ RNA	15.1$_{+2.2}$	37.5$_{+0.8}$	41.0$_{-0.5}$	12.2$_{+1.3}$	19.8$_{-5.4}$	23.4$_{+1.8}$
RSV	Original	17.4	36.7	42.5	8.2	25.4	21.4
	w/ RNA	16.9$_{-0.5}$	37.5$_{+0.8}$	42.8$_{+0.3}$	10.4$_{+2.2}$	23.2$_{-2.2}$	25.6$_{+4.2}$
Preference Optimization							
NPO+KL	Original	14.2	36.2	40.1	10.4	27.0	21.6
	w/ RNA	14.7$_{+0.5}$	36.7$_{+0.5}$	38.9$_{-1.2}$	9.3$_{-1.1}$	22.7$_{-4.3}$	23.6$_{+2.0}$
NPO+MSE	Original	10.6	37.6	41.0	11.3	26.1	22.3
	w/ RNA	11.2$_{+0.6}$	35.7$_{-1.9}$	38.8$_{-2.2}$	9.1$_{-2.2}$	23.4$_{-2.7}$	21.4$_{-0.9}$
DPO+KL	Original	11.9	36.1	37.2	11.3	23.2	19.8
	w/ RNA	11.3$_{-0.6}$	36.9$_{+0.8}$	38.7$_{+1.5}$	11.3$_{+0.0}$	23.2$_{+0.0}$	19.8$_{+0.0}$
DPO+MSE	Original	10.0	36.0	39.8	11.6	23.8	22.5
	w/ RNA	14.9$_{+4.9}$	37.5$_{+1.5}$	40.5$_{+0.7}$	14.2$_{+2.6}$	24.3$_{+0.5}$	24.1$_{+1.6}$
SimNPO+KL	Original	15.6	36.5	41.0	11.1	20.9	18.7
	w/ RNA	17.8$_{+2.2}$	37.5$_{+1.0}$	41.8$_{+0.8}$	8.0$_{-3.1}$	23.6$_{+2.7}$	20.3$_{+1.6}$
SimNPO+MSE	Original	11.0	38.2	39.5	8.2	24.5	24.3
	w/ RNA	11.0$_{+0.0}$	37.9$_{-0.3}$	40.2$_{+0.7}$	13.6$_{+5.4}$	25.6$_{+1.1}$	23.2$_{-1.1}$

I Performance of Other Models

Our experiments in the main text are based on the Zephyr-7B model, which serves as a representative setup. To assess RNA's generalization beyond the original setup, we conducted additional experiments using the Llama-3-8B (Dubey et al., 2024) and Mistral-7B (Jiang et al., 2023) models on two representative unlearning methods from 2 classes: RMU and NPO+KL, across various tasks. These results provide further empirical evidence for RNA's generalization and robustness.

Hyperparameters. All models are fine-tuned using Adam (Kingma, 2014) for $T = 500$ update steps with a learning rate of 5×10^{-5}, batch size 4, and maximum sequence length of 500 for WMDP-Biology and 768 for WMDP-Cyber. The unlearned layer is fixed at $l = 7$. Retain weights are set to $\alpha_{\text{biology}} = \alpha_{\text{cyber}} = 1200$ for both models. The coefficient values are $c_{\text{biology}} = c_{\text{cyber}} = 20$ for Llama-3-8B and $c_{\text{biology}} = c_{\text{cyber}} = 6.5$ for Mistral-7B. For NPO+KL, we perform a grid search over $(\alpha_{\text{biology}}, \alpha_{\text{cyber}})$ and select $(5, 10)$ for Llama-3-8B and $(30, 40)$ for Mistral-7B. For RMU w/ RNA, we set the perturbed layer to $l = 7$, tune the noise scale via grid search, and report the best performance at $\nu = 7 \times 10^{-2}$ (Llama-3-8B) and $\nu = 3 \times 10^{-2}$ (Mistral-7B). For NPO+KL w/ RNA, we also perturb layer $l = 7$ and select the best scales $\nu = 6 \times 10^{-2}$ (Llama-3-8B) and $\nu = 3 \times 10^{-3}$ (Mistral-7B).

Results. As shown in Table 12, across Llama-3-8B and Mistral-7B, RNA significantly enhances unlearning robustness of models while introducing a small trade-off on forget performance. For forget-tasks (WMDP-Biology and Cyber), RNA slightly increases the accuracy, *e.g.,* RMU on Llama-3-8B drops from 31.4 → 34.6 (−3.2) and NPO+KL from 27.9 → 33.2 (−5.3). For retain-tasks (MMLU and Perturbed MMLU and MMLU subsets), RNA substantially improves performance, particularly on perturbed MMLU and MMLU subsets (C. Bio. and C. Sec.). For example, perturbed MMLU on Llama-3-8B with NPO+KL improves from 26.2 → 47.3 (+21.1), and MMLU C. Bio. from 30.5 → 55.5 (+25.0). Overall, RNA effectively recovers or enhances accuracy on retain-tasks while slightly compromising forget-task performance, demonstrating a favorable trade-off between unlearning and model robustness.

Table 12: Performance of Llama-3-8B and Mistral-7B on WMDP and MMLU, Perturbed MMLU, MMLU subsets benchmarks using RMU and NPO+KL, comparing Original vs. **w/ RNA**. Improvements are shown in blue, drops in red.

Models & Methods	Llama-3-8B				Mistral-7B			
	RMU		NPO+KL		RMU		NPO+KL	
	Original	**w/ RNA**	Original	**w/ RNA**	Original	**w/ RNA**	Original	**w/ RNA**
WMDP (↓)	31.4	34.6$_{-3.2}$	27.9	33.2$_{-5.3}$	31.7	31.7$_{+0.0}$	29.3	34.0$_{-4.7}$
MMLU (↑)	60.3	60.2$_{-0.1}$	53.8	54.4$_{+0.6}$	58.2	58.6$_{+0.4}$	56.5	56.4$_{-0.1}$
Perturbed MMLU (↑)	34.4	47.3$_{+12.9}$	26.2	47.3$_{+21.1}$	27.2	42.2$_{+15.0}$	31.4	53.5$_{+22.1}$
MMLU C. Bio. (↑)	34.7	60.4$_{+25.7}$	30.5	55.5$_{+25.0}$	25.0	38.1$_{+13.1}$	32.6	52.0$_{+19.4}$
MMLU C. Sec. (↑)	29.0	33.0$_{+4.0}$	30.0	46.0$_{+16.0}$	33.0	46.0$_{+13.0}$	35.0	30.0$_{-5.0}$

J Performance of RNA under Miscalibrated Unlearning

Miscalibrated unlearning refers to scenarios where unlearning is either **over-unlearn**, *i.e.,* the model successfully unlearns the target knowledge but suffers catastrophic degradation in general knowledge, or **under-unlearn**, *i.e.,* the model fails to sufficiently remove the target knowledge. When unlearning is under-unlearn, the backdoor signals are too weak, *i.e.,* forget-representations are not well-aligned with random vectors, making them less harmful when they appear in retain-queries. Over-unlearn occurs when the unlearning methods fail to distinguish between forget and retain knowledge, leading to catastrophic degradation of both. In such cases, random noises injected by RNA may be either redundant (for small ν) or recover both forget and retain knowledge (for large ν). Theoretically, RNA is a variance reduction defense against sensitivity caused by forget-tokens, **not a method for miscalibrated unlearning strength**. When unlearning is poorly calibrated, smoothing from RNA becomes ineffective. We conduct an empirical analysis of these two cases to evaluate our intuition. Results are shown in Figure 13 and Figure 14. Overall, we found that RNA fails to enhance retain-robustness when unlearning is poorly calibrated.

Setup. We employ MUSE-NEWS_TARGET[1] as the base model for unlearning. MUSE-NEWS_TARGET is the Llama-2-7B (Touvron et al., 2023) model fine-tuned on the News corpus (BBC news articles). We employ two representative unlearning methods, RMU and NPO+KL.

Hyperparameters. For RMU, we perform a heuristic search over the coefficient $c \in [100, 120, 130, 140, 150]$. We set the retain-weight $\alpha_r = 1200$ (coefficient of forget-loss), forget-weight $\alpha_f = 1.0$ (coefficient of retain-loss), $T = 500$ gradient steps, unlearn with layer $l = 7$, RNA noise added at layer 7, learning rate $2e-5$, maximum sequence length 256. MUSE-News forget-set is used as $\mathcal{D}_f$, Wikitext is used as $\mathcal{D}_r$. For NPO+KL, we search over $(\alpha_f, \alpha_r) \in [(1,1), (5,1), (10,1), (20,1), (1,5), (1,10), (1,20)]$, β is set to 0.1. For each pair (α_f, α_r), we grid search RNA's noise scale $\nu \in [1e-3, 2e-3, 3e-3]$. We report VerbMem and KnowMem.

K Limitations

We posit the following limitations of this study and discuss potential future works.

We have evaluated our methods primarily on WMDP, a widely used and representative benchmark. We acknowledge the existence of other benchmarks, such as MUSE (Shi et al., 2025) and TOFU (Maini et al., 2024), but these are less suitable for our experimental setup. Specifically, TOFU is designed to remove the influence of specific data points, making it less applicable in generative settings. While MUSE could be suitable, previous work (Shi et al., 2025) has shown that methods evaluated on MUSE often exhibit over-forgetting or under-forgetting. Since our study focuses on retain-robustness, which requires a careful balance between forgetting and retaining, MUSE is not ideal. These factors make it challenging to apply MUSE and TOFU in our current experiments.

[1] https://huggingface.co/muse-bench/MUSE-news_target

Table 13: Performance of original RMU unlearned models and RNA models under over-unlearn and under-unlearn in MUSE News.

Method		$\textbf{VerbMem}_f \downarrow$	$\textbf{KnowMem}_f \downarrow$	$\textbf{KnowMem}_r$ (benign) $\uparrow$	$\textbf{KnowMem}_r$ (perturbed) $\uparrow$
Base model (MUSE-news_target)		57.2	64.2	64.2	51.8
RMU ($c = 100$)	Original	57.3	65.5	55.0	51.0 (**under-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	56.7	66.1	56.0	49.3
	w/ RNA ($\nu = 2\text{e}{-3}$)	56.6	65.7	55.1	50.8
	w/ RNA ($\nu = 3\text{e}{-3}$)	56.7	66.1	55.8	50.4
RMU ($c = 110$)	Original	56.5	66.1	55.1	50.2 (**under-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	56.4	66.1	55.7	50.7
	w/ RNA ($\nu = 2\text{e}{-3}$)	56.6	64.3	54.9	49.8
	w/ RNA ($\nu = 3\text{e}{-3}$)	55.3	65.0	54.9	50.8
RMU ($c = 120$)	Original	56.2	64.1	55.8	49.6 (**under-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	55.8	65.9	55.2	50.1
	w/ RNA ($\nu = 2\text{e}{-3}$)	55.9	66.3	55.3	50.2
	w/ RNA ($\nu = 3\text{e}{-3}$)	56.0	66.1	55.9	50.8
RMU ($c = 130$)	Original	54.1	56.2	49.0	45.4 (**under-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	55.7	65.2	56.3	49.8
	w/ RNA ($\nu = 2\text{e}{-3}$)	55.9	65.2	55.9	50.0
	w/ RNA ($\nu = 3\text{e}{-3}$)	55.0	66.0	56.0	50.4
RMU ($c = 140$)	Original	53.9	43.2	36.3	38.1 (**under-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	54.6	62.9	51.9	48.2
	w/ RNA ($\nu = 2\text{e}{-3}$)	55.0	62.8	52.8	49.1
	w/ RNA ($\nu = 3\text{e}{-3}$)	55.7	64.9	54.5	50.2
RMU ($c = 150$)	Original	49.2	13.7	18.0	18.7 (**over-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	53.6	48.6	43.5	37.8
	w/ RNA ($\nu = 2\text{e}{-3}$)	54.2	51.4	44.2	39.6
	w/ RNA ($\nu = 3\text{e}{-3}$)	54.3	53.9	51.0	45.3

Due to computational constraints, experiments are conducted only on the 7B or 8B models and with updates to a limited set of layer parameters, which may risk overlooking interesting aspects of generalization. Although RNA has demonstrated effectiveness, it relies heavily on hyperparameter grid search to identify an optimal noise scale, making it computationally expensive for extremely large models with hundreds of billions of parameters.

L AI Usage Declaration

AI tools were used for grammar checking and formatting of tables and figures. To our best knowledge and belief, we hereby declare that, all technical content and implementations were written by the authors.

Table 14: Performance of NPO+KL unlearned models and NPO+KL w/ RNA models under over-unlearn and under-unlearn in MUSE News.

Method		$\textbf{VerbMem}_f \downarrow$	$\textbf{KnowMem}_f \downarrow$	$\textbf{KnowMem}_r$ (benign) $\uparrow$	$\textbf{KnowMem}_r$ (perturbed) $\uparrow$
Base model (MUSE-news_target)		57.2	64.2	64.2	51.8
NPO+KL $(1,1)$	Original	57.5	61.4	52.6	48.8 (**under-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	56.8	61.0	52.0	47.6
	w/ RNA ($\nu = 2\text{e}{-3}$)	56.8	61.0	52.8	46.6
	w/ RNA ($\nu = 3\text{e}{-3}$)	56.5	60.2	52.7	46.5
NPO+KL $(1,5)$	Original	57.5	59.5	52.7	47.8 (**under-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	58.0	59.9	53.8	47.0
	w/ RNA ($\nu = 2\text{e}{-3}$)	56.3	60.4	53.6	46.8
	w/ RNA ($\nu = 3\text{e}{-3}$)	57.5	59.8	53.0	46.7
NPO+KL $(1,10)$	Original	58.6	60.0	52.6	45.7 (**under-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	58.7	59.8	53.0	46.4
	w/ RNA ($\nu = 2\text{e}{-3}$)	58.3	59.9	53.0	46.7
	w/ RNA ($\nu = 3\text{e}{-3}$)	58.9	60.1	53.9	46.5
NPO+KL $(1,20)$	Original	57.7	59.5	54.3	47.4 (**under-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	58.2	61.2	53.8	47.6
	w/ RNA ($\nu = 2\text{e}{-3}$)	58.8	62.8	53.5	46.5
	w/ RNA ($\nu = 3\text{e}{-3}$)	58.2	62.8	54.7	47.7
NPO+KL $(5,1)$	Original	56.8	59.9	51.8	48.6 (**under-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	55.7	60.0	52.6	47.8
	w/ RNA ($\nu = 2\text{e}{-3}$)	56.9	59.4	53.1	48.5
	w/ RNA ($\nu = 3\text{e}{-3}$)	56.7	60.1	51.9	47.7
NPO+KL $(10,1)$	Original	57.3	60.6	53.1	48.4 (**under-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	56.7	61.5	52.3	48.2
	w/ RNA ($\nu = 2\text{e}{-3}$)	55.7	59.9	52.1	48.1
	w/ RNA ($\nu = 3\text{e}{-3}$)	57.1	59.3	52.6	48.3
NPO+KL $(20,1)$	Original	56.4	58.2	52.1	48.2 (**under-unlearn**)
	w/ RNA ($\nu = 1\text{e}{-3}$)	56.6	59.3	52.1	47.8
	w/ RNA ($\nu = 2\text{e}{-3}$)	56.6	60.6	51.9	48.1
	w/ RNA ($\nu = 3\text{e}{-3}$)	56.1	60.8	51.7	47.9