# Improving LLM Unlearning Robustness via Random Perturbations

**Dang Huu-Tien**[†*] **Hoang Thanh-Tung**[◇] **Anh Bui**[♣]

**Le-Minh Nguyen**[†] **Naoya Inoue**[†,‡]

## Abstract

In this paper, we show that current state-of-the-art LLM unlearning methods inherently reduce models' robustness, causing them to misbehave even when a *single non-adversarial* forget-token is in the retain-query. Toward understanding underlying causes, we reframe the unlearning process as backdoor attacks and defenses: forget-tokens act as backdoor triggers that, when activated in retain-queries, cause disruptions in unlearned models' behaviors, similar to successful backdoor attacks. To mitigate this vulnerability, we propose Random Noise Augmentation (RNA)—a plug-and-play, 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, maintains unlearning performances while introducing no additional computational overhead.

## 1 Introduction

Modern LLMs are pre-trained on massive text corpora and then post-trained with Reinforcement Learning from Human Feedback (RLHF; [9, 68, 54, 41]) to be helpful and harmless [2]. Recent studies have shown that despite safety enhancements, aligned LLMs can still exhibit harmful and undesirable behaviors, such as generating toxic content [62], producing copyrighted material [26, 14, 61], gender bias [4], leaking sensitive and private information [39, 43], and potentially aiding malicious uses such as cyberattacks, chemical attacks, and bioweapons development [17, 49, 30]. As LLMs advance in size and capabilities at an unprecedented speed, concerns about their potential risks continue to grow.

Machine Unlearning (MU; [7, 5, 40, 63, 10]) is an approach aiming to *robustly* (1) remove or suppress 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 [30, 12, 34, 32], sequential unlearning [52], target relearning attacks [21], removing/steering specific directions in the latent space [34, 50], quantization [67], or even simply fine-tuning on unrelated tasks [13, 34].

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. [57], who examined the robustness of Representation Misdirection for Unlearning (RMU; [30]), demonstrating that RMU-unlearned models are fragile when asked with retain-queries (*e.g.*, Q&A about general

[*]Correspondence to: dtienuet@gmail.com; [†]Japan Advanced Institute of Science and Technology; [◇]VNU University of Engineering and Technology, Vietnam; [♣]Monash University [‡]RIKEN.
Our implementation is available at `https://github.com/RebelsNLU-jaist/llmu-robustness`

Preprint.

knowledge) containing forget-tokens (tokens in the forget-set). However, many critical questions remain unanswered. In this paper, we make the following contributions:

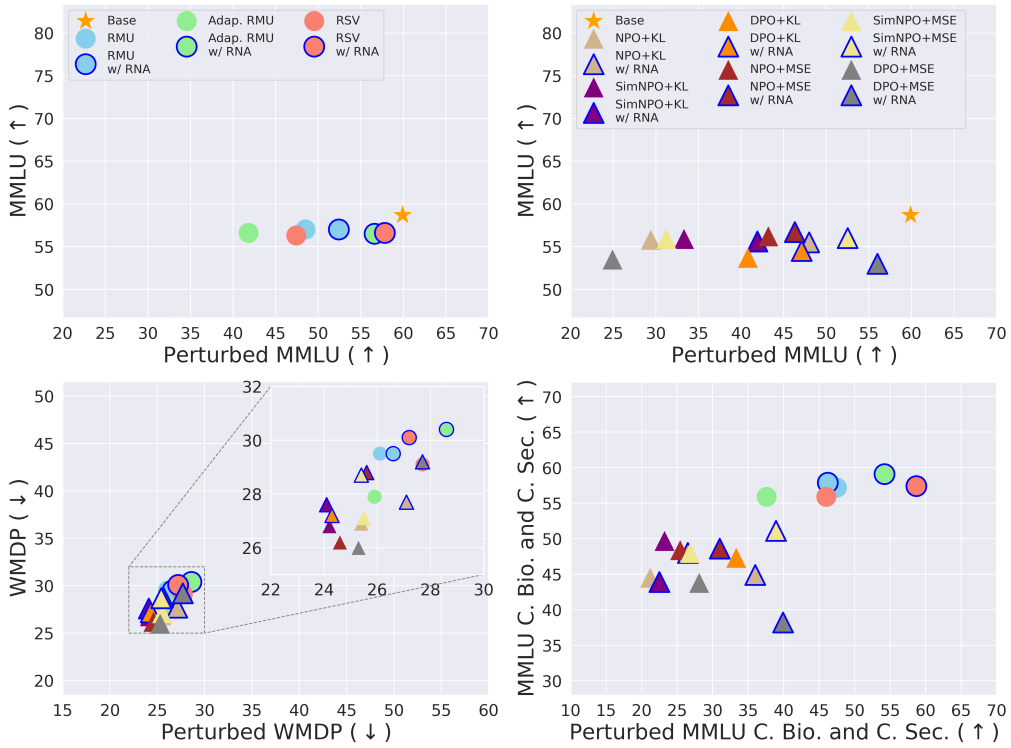


Figure 1: **Top-left**: Accuracy of RM and RM w/ RNA models on MMLU [20] and perturbed MMLU (MMLU QA contains forget-tokens; see Appendix A for details). **Top-right**: Accuracy of PO and PO w/ RNA models on MMLU and perturbed MMLU. **Bottom-left**: Accuracy of all unlearned models on WMDP [30] and perturbed WMDP. **Bottom-right**: Accuracy of all unlearned models on MMLU 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.

① 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 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 the unlearning methods inadvertently learn to align forget-tokens (backdoor triggers) with the target representations (the 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 and introduce *Random Noise Augmentation (RNA)*, a plug-and-play, model- and method-agnostic approach which adds small, independent Gaussian noise to each retain-query's representation during fine-tuning to reduce the model's sensitivity to forget-tokens.

③ Through theoretical and empirical analysis, we show that RNA significantly improves the robustness of unlearned models while maintaining original forget and retain performances (Figure 1).

## 2 Related Works and Preliminaries

### 2.1 Related Works

**Machine Unlearning.** MU has become one of the most important tools for ensuring the safety and protecting the privacy of LLMs [63, 40]. Most recent works on MU focus on developing algorithms for different tasks, domains, and settings [44, 58, 25, 52, 8, 42, 38, 60, 28], 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 [30, 35, 3, 32], sequential unlearning [52], quantization [67], fine-tuning unlearned models on unrelated tasks [13, 34], and adversarial attacks [21, 64, 53, 22] and developed methods for improving forget-robustness of MU algorithms [51, 56, 55, 16, 67]. This paper explores the "retain-robustness" of MU algorithms, studying the robustness of MU algorithms in robustly retaining the original model's general knowledge and capabilities. Thaker et al. [57] presented preliminary results showing that state-of-the-art MU algorithms do not preserve the original model's knowledge and capabilities. We bridge the gap in retain-robustness research by introducing RNA, a simple data augmentation method inspired by adversarial training to improve the robustness of MU algorithms.

### 2.2 Preliminaries

**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 $f_\theta$ be a model parameterized by $\theta$, and $\ell(y|x;\theta)$ is the loss of input $x$ with respect to a target output $y$ in model $f_\theta$. A commonly used form of unlearning involves minimizing the following two-part loss:

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

where $y^f$, $y^r$ are the target outputs of forget and retain input, respectively, $\alpha \in \mathbb{R}^+$ is a retain weight. We consider two widely used classes of LLM unlearning methods, which rely on Representation Misdirection (RM) and Preference Optimization (PO).

#### 2.2.1 Representation Misdirection

Representation Misdirection (RMU and variants) are unlearning methods that achieve unlearning by manipulating latent representations during fine-tuning. Denote $\mathbf{z}^f_\theta, \mathbf{z}^r_\theta \in \mathbb{R}^{n\times d_l}$ the output hidden state of $n$-tokens in forget-sample $x^f$ and in retain-sample $x^r$, respectively, at layer $l$ in model $f_\theta$, where $d_l$ is the dimension of hidden states at layer $l$. In this context, the activations from the MLP module in the transformer layer is used as the latent representation.

**Representation Misdirection for Unlearning (RMU;** [30]**)** pushes the latent representation of forget-tokens to a predetermined random representation $y^f = c\mathbf{u}$, 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:

$$\mathcal{L}^{\text{RMU}} = \mathbb{E}_{x^f\sim\mathcal{D}_f}||\mathbf{z}^f_\theta - c\mathbf{u}||_2^2 + \alpha\mathbb{E}_{x^r\sim\mathcal{D}_r}||\mathbf{z}^r_\theta - \mathbf{z}^r_{\theta^{\text{ref}}}||_2^2, \tag{2}$$

where $\theta$ and $\theta^{\text{ref}}$ are the parameters of the updated and reference (frozen weight) models, respectively, and $\alpha \in \mathbb{R}^+$ is a retain weight.

**Adaptive RMU** [11] is a variant of RMU that adaptively changes the coefficient of random vector $\mathbf{u}$ in the forget-loss based on the norm of forget-sample in the reference model. The target random representation $y^f = \beta||\mathbf{z}^f_{\theta^{\text{ref}}}||_2\mathbf{u}$, where $\beta \in \mathbb{R}^+$ is a scaling factor.

**Random Steering Vector (RSV).** Additionally, we implement RSV—a variant of RMU that uses the target random representation $y^f = \mathbf{z}^f_{\theta^{\text{ref}}} + c\boldsymbol{\epsilon}$, 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; [66])** treats forget-samples as negative preference samples in Direct Preference Optimization framework (DPO; [46]). NPO can be viewed as a gradient ascent variant with adaptive gradient weights that allows more controlled and stable optimization:

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

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 $\pi_\theta(y^f|x^f)$, $\pi_{\theta^{\text{ref}}}(y^f|x^f)$ denotes the predicted probability of $y^f$ given $x^f$ in the model $f_\theta$ and reference model $f_{\theta^{\text{ref}}}$ (frozen weight) respectively.

**Simple Negative Preference Optimization (SimNPO; [15])** simplifies NPO by using a normalized sequence log-probability and introducing a reward margin hyperparameter $\gamma\geq 0$:

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

where $|y^f|$ is the length of output sequence $y^f$.

**Direct Preference Optimization (DPO; [46]).** As a baseline, Zhang et al. [66], Maini et al. [36], Yuan et al. [65]) adopted standard DPO, using a refusal answer $y^{\text{idk}}\in\mathcal{D}_{\text{idk}}$ such as "I Don't Know" as the positive samples and forget-samples as negative samples.

To preserve general capabilities, we use Mean Squared Error (MSE): $\mathcal{L}^{\text{MSE}} = \mathbb{E}_{(x^r,y^r)\sim\mathcal{D}_r}||\log\pi_\theta(x^r) - \log\pi_{\theta^{\text{ref}}}(x^r)||_2^2$ or Kullback–Leibler divergence (KL) $\mathcal{L}^{\text{KL}} = \mathbb{E}_{(x^r,y^r)\sim\mathcal{D}_r}\text{KL}\left(\log\pi_\theta(x^r),\log\pi_{\theta^{\text{ref}}}(x^r)\right)$ 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 $\mathbf{z}_\theta^f+\mathbf{v}$ be the steered latent representation of forget-sample $x^f$ in $f_\theta$ 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 notation of the GLVM, that is, GLVM $f_\theta$ generates target output $y^f$ given the latent variable $\mathbf{z}_\theta^f$. Let $\ell(y^f|\mathbf{z}_\theta^f+\mathbf{v};\theta)$ be the loss of generating $y^f$ given $\mathbf{z}_\theta^f+\mathbf{v}$ in model $f_\theta$. For simplicity, we write $\ell(y^f|\mathbf{z}_\theta^f+\mathbf{v})$ to present $\ell(y^f|\mathbf{z}_\theta^f+\mathbf{v};\theta)$. Following Koh and Liang [27], we assume that the loss is twice-differentiable and locally convex. Since $\mathbf{v}$ is small, we approximate the function $\ell(y^f|\mathbf{z}_\theta^f+\mathbf{v})$ using the second-order Taylor approximation:

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

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

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

A classic result from Hutchinson [24] tell us that $\mathbb{E}_{\mathbf{v}}[\mathbf{v}^\top\nabla^2_{\mathbf{z}_\theta^f}\ell(y^f|\mathbf{z}_\theta^f)\mathbf{v}] = \mu\text{Tr}(\nabla^2_{\mathbf{z}_\theta^f}\ell(y^f|\mathbf{z}_\theta^f))$, where $\text{Tr}(\nabla^2_{\mathbf{z}_\theta^f}\ell(y^f|\mathbf{z}_\theta^f))>0$ is the trace of the positive definite Hessian matrix $\nabla^2_{\mathbf{z}_\theta^f}\ell(y^f|\mathbf{z}_\theta^f)$ (by assumption). Since $\mu\in\mathbb{R}^+$, the loss of generating $y^f$ given latent variable $\mathbf{z}_\theta^f$ is *increases*, that is,

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

While presenting in different formulations, PO and RM *share a common high-level principle—maximizing the loss of forget-samples* $\ell(y^f|\mathbf{z}_\theta^f)$. 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.

# 4 Analysis on the Robustness of Unlearned Models

## 4.1 Threat Model

We 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.*

**Unlearning guarantee.** The 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 $x_i^r$ be the generated token given the previous retain sequence $x_{<i}^r$ in the unlearned model $f^u$. $x_{<i}^{r,\text{per}}$ denotes the perturbed retain-query (the retain-query that contains forget-tokens). Let $\mathbf{z}_{<i}^r$ and $\mathbf{z}_{<i}^{r,\text{per}}$ be the representation of $x_{<i}^r$ and $x_{<i}^{r,\text{per}}$ respectively obtained from $f^u$ at layer $l$. To formalize the analysis, we introduce the following assumption.

**Assumption 4.1.** The latent representation of perturbed retain-query in the unlearned model 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 4.1 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 $\eta$ controls the magnitude of perturbations capturing the variation of forget-tokens can be appeared in the perturbed retain-queries. Based on this assumption, we derive the change in the output representation of the generated tokens as follows.

**Theorem 4.2.** *If Assumption 4.1 holds, the change in the output representation of the generated token* $x_i^r$ *given the perturbed retain-query* $x_{<i}^{r,\text{per}}$ *and the benign retain-query* $x_{<i}^r$ *in the unlearned model* $f^u$*, defined as* $\Delta = f^u(x_i^r|x_{<i}^{r,per}) - f^u(x_i^r|x_{<i}^r)$*, follows the Normal distribution* $\mathcal{N}(\mathbf{0}, \eta\boldsymbol{J}^\top\boldsymbol{J})$*, where* $\boldsymbol{J} = \nabla_{\mathbf{z}_{<i}^r} f^u(x_i^r|x_{<i}^r)$ *is the Jacobian of* $f^u(x_i^r|x_{<i}^r)$ *with respect to* $\mathbf{z}_{<i}^r$.

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

Theorem 4.2 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 $\Delta$ is determined by the product of $\eta$ and $\boldsymbol{J}^\top\boldsymbol{J}$, where $\eta$ is the scalar coefficient controlling the magnitude of the added noise $\boldsymbol{\epsilon}$ in Assumption 4.1, 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 $\Delta$ is challenging. However, a larger $\eta$ amplifies the variance of $\Delta$, thereby increasing the randomness in the output. This suggests the following empirical analysis: (1) forget-tokens with the larger representation randomness tend to induce more variability in the predictions. (2) In RM forget-losses,

a larger magnitude of the target random vector further increases the randomness of the forget-token representation, *i.e., the larger coefficient $c$ (or $\beta$), the less robustness of the RM unlearned models.* In Section 6, we present an empirical analysis to validate the analysis.

# 5 Random Noise Augmentation

## 5.1 Motivation: Unlearning as a Backdoor Attack and Defense Problem

As a motivation, our first start is an assumption that the unlearning process can be decomposed into two parts: the "forgetting" process and the "retaining" process. We then reframe the *unlearning as a backdoor attack and defense problem* and propose a simple yet effective solution to enhance the robustness of unlearned models.

**"Forgetting" as a backdoor attack.** Consider the supervised learning setting with the objective of learning a model $f_\theta : \mathcal{X} \mapsto \mathcal{Y}$. 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 $\mathcal{Z}_f = \{(\mathbf{z}^f_\theta, \mathbf{z}^f_{\theta^{\text{ref}}})\}_i$, where $\mathbf{z}^f_\theta \in \mathcal{X}$ is the input, $\mathbf{z}^f_{\theta^{\text{ref}}} \in \mathcal{Y}$ is the target output, and a retain-set $\mathcal{Z}_r = \{(\mathbf{z}^r_\theta, \mathbf{z}^r_{\theta^{\text{ref}}})\}_j$ where $\mathbf{z}^r_\theta \in \mathcal{X}$ and $\mathbf{z}^r_{\theta^{\text{ref}}} \in \mathcal{Y}$. Each forget-sample $(\mathbf{z}^f_\theta, \mathbf{z}^f_{\theta^{\text{ref}}})$ is transformed into a backdoor-sample $(T(\mathbf{z}^f_\theta), \Omega(\mathbf{z}^f_{\theta^{\text{ref}}}))$, where $\Omega$ 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 $\Omega$ specifies the behavior of the model when the backdoor trigger is activated. In the "forgetting", $T$ is an identity function *i.e.* $T(\mathbf{z}^f_\theta) = \mathbf{z}^f_\theta$ and $\Omega$ is a function that maps $\mathbf{z}^f_{\theta^{\text{ref}}}$ to the adversarial-perturbed representation (*e.g.*, $c\mathbf{u}$ in RMU). We train model $f_\theta$ with "poisoned" forget-set $\mathcal{Z}^{\text{poisoned}}_f = \{(T(\mathbf{z}^f_\theta)), \Omega(\mathbf{z}^f_{\theta^{\text{ref}}}))\}_i$ and benign retain-set $\mathcal{Z}_r = \{(\mathbf{z}^r_\theta, \mathbf{z}^r_{\theta^{\text{ref}}})\}_j$, as follows:

$$\theta^* = \arg\min_\theta \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{Z}} \left[\mathcal{L}(f_\theta(\mathbf{x}), \mathbf{y})\right], \tag{10}$$

where $\mathbf{x}$ is either $\mathbf{z}^f_\theta$ or $\mathbf{z}^r_\theta$ and $\mathbf{y}$ is either $\mathbf{z}^f_{\theta^{\text{ref}}}$ or $\mathbf{z}^r_{\theta^{\text{ref}}}$. During inference, for a retain-input $\mathbf{z}^r_\theta$ and forget-input $\mathbf{z}^f_\theta$ the unlearned model should behave as follows:

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

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

This formulation suggests that ***current state-of-the-art 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 the activation of the backdoor trigger in these queries, leading the model to misbehave.*** *This backdoor explanation further highlights the evidence that current LLM unlearning methods do not truly erase knowledge; in fact, they intentionally decide that the model's target knowledge/behaviors should not be surfaced* [29, 10].

**"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 forget-tokens. Inspired by previous works using random noise defenses against adversarial attacks [31, 19, 48, 45, 6, 23], we propose Random Noise Augmentation (RNA), adds a small, independent random Gaussian noise $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \nu\boldsymbol{I})$, $\nu \in \mathbb{R}^+$ to each *retain-representation in the reference model* during fine-tuning.

**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 plug-and-play, model- and method-agnostic: RNA can be applied to any deep networks and generalizes to the 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.

**Algorithm 1** Random Noise Augmentation

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

**Ensure:** return logit and representation of $x^r$.

```
Sample a random vector δ ~ N(0, νI).
for layer ∈ [1...L] do
    if layer == l then
        z^r_{θ^ref} ← z^r_{θ^ref} + δ.
    end if
end for
return (logit^r_{θ^ref}, z^r_{θ^ref})
```

(2) RNA modifies only a single layer's representation without requiring extra forward passes or gradient computations, making it scalable and efficient. See Appendix E for an ablation study on effects of applying RNA to different latent spaces. (3) RNA is theoretically guaranteed (Section 5.2).

### 5.2 Robustness of RNA Models

**Assumption 5.1.** The latent representation of the retain-query $x^r_{<i}$ is randomized in the RNA model, that is, $\mathbf{z}^r_{\theta^{\text{rna}}} = \mathbf{z}^r_{\theta^u} + \boldsymbol{\delta}$, 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 $f^{\text{rna}}$ the RNA model, $f^u$ the unlearned model, and $\mathcal{J}(.,.)$ be a loss function. We consider the change in the loss of the generated token $x^r_i$ given the perturbed retain-query and the retain-query in the unlearned model $f^{\text{u}}$: $\Delta\mathcal{J}^u = \mathcal{J}(f^u(x^r_i|x^{r,\text{per}}_{<i})) - \mathcal{J}(f^u(x^r_i|x^r_{<i}))$. Since the predicted output $f^u(x^r_i|x^{r,\text{per}}_{<i})$ is randomized (c.f. Theorem 4.2), the loss is increased, resulting in $\Delta\mathcal{J}^u > 0$.

The change in the loss in RNA model $f^{\text{rna}}$ is $\Delta\mathcal{J}^{\text{rna}} = \mathcal{J}(f^{\text{rna}}(x^r_i|x^{r,\text{per}}_{<i})) - \mathcal{J}(f^{\text{rna}}(x^r_i|x^r_{<i}))$. If $f^{\text{rna}}$ 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 5.2.** *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 4.1 and Assumption 5.1 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 $\frac{1}{2} - \frac{1}{\pi}\arctan\left[\sqrt{\frac{\eta}{\nu}}\left(1 + \frac{||\boldsymbol{g}^{per}||_2}{||\boldsymbol{g}||_2}\right)^{-1}\right]$, where $\boldsymbol{g}^{per} = \nabla_{\mathbf{z}^{r,per}_{<i}}\mathcal{J}(f^u(x^r_i|x^{r,per}_{<i}))$ and $\boldsymbol{g} = \nabla_{\mathbf{z}^r_{<i}}\mathcal{J}(f^u(x^r_i|x^r_{<i}))$ are the gradients of the loss of generated token $x^r_i$ with respect to $\mathbf{z}^{r,per}_{<i}$ and $\mathbf{z}^r_{<i}$.*

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

Theorem 5.2 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 $\arctan\left[\sqrt{\frac{\eta}{\nu}}\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||_2}{||\boldsymbol{g}||_2}\right)^{-1}\right]$. Since $\arctan$ is monotonically increasing, the robustness of unlearned models increases as $\sqrt{\frac{\eta}{\nu}}\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||_2}{||\boldsymbol{g}||_2}\right)^{-1}$ decreases. The product $\sqrt{\frac{\eta}{\nu}}\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||_2}{||\boldsymbol{g}||_2}\right)^{-1}$ is characterized by two terms: the root of the ratio $\frac{\eta}{\nu}$ and $\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||_2}{||\boldsymbol{g}||_2}\right)^{-1}$. First, if $\nu$ and $\eta$ are fixed, a larger ratio $\frac{||\boldsymbol{g}^{\text{per}}||_2}{||\boldsymbol{g}||_2}$ means a smaller $\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||_2}{||\boldsymbol{g}||_2}\right)^{-1}$, that is, a more robustness of the unlearned models. However, searching for all input and analyzing the effects of $\frac{||\boldsymbol{g}^{\text{per}}||_2}{||\boldsymbol{g}||_2}$ would be challenging due to the input-dependent property of $\boldsymbol{g}$ and $\boldsymbol{g}^{\text{per}}$. Second, consider the effect of $\frac{\eta}{\nu}$. If $\eta$ is fixed (the magnitude of the noise caused by forget-tokens), *the larger $\nu$ is, the more robust the unlearned model becomes*. However, since the probability is bounded, the robustness of unlearned models reaches *a saturation point* as $\nu$ increases. We present an empirical analysis in Section 6 to validate the claims.

## 6 Empirical Analysis

### 6.1 Experimental Setup

**Models and datasets.** We conduct our experiments using Zephyr-7B-$\beta$ [59]. For fine-tuning, we use the WMDP Biology and WMDP Cyber forget-sets as $\mathcal{D}_f$, and Wikitext [37] as the retain-set $\mathcal{D}_r$. For evaluation, we use the WMDP Biology and 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 drawn from the forget-set. Following prior work [57], we use "SARS-CoV-2," a frequent term in the WMDP Biology forget-set. See Appendix A.2 for details of the prompt templates.

**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 present key results supporting our theoretical analysis in the main text, and defer the full experimental setup and additional results to the Appendix A.4.

### 6.2 Main Results and Analysis

**RNA improves unlearned models' robustness while preserving their original forget/retain performances.** Figure 1 (top-left and top-right) 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 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 bottom-right).

**Trade-off between the coefficient in RM and robustness.** As suggested by Theorem 4.2, increasing either the coefficient $c$ or scaling factor $\beta$ is expected to reduce the unlearned model's robustness. To

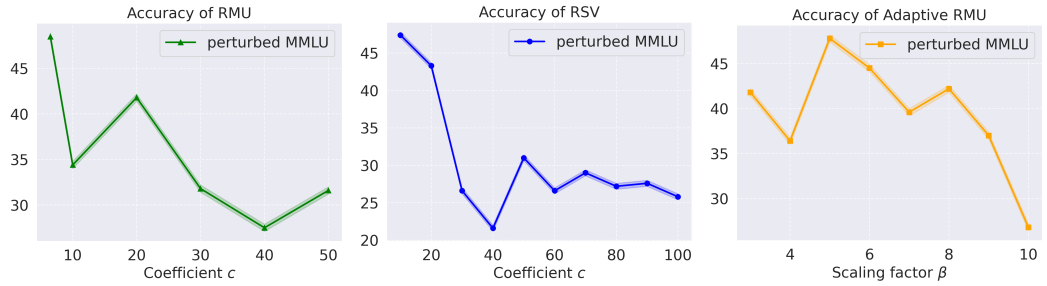


Figure 2: Accuracy of RM models on perturbed MMLU across values of coefficient $c$ and scaling factor $\beta$. The accuracy tends to decrease as either $c$ or $\beta$ increases.

validate this claim, we fix the unlearn layer at $l = 7$ and grid search over values of $c$ and $\beta$, 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 $\beta$ increases. Previous works [30, 11] performed grid search for $c$ and $\beta$, selecting values that yielded optimal accuracy and observed that deeper unlearn layers require larger values of $c$ (or $\beta$) to achieve effective unlearning. However, our results demonstrate that increasing the coefficient $c$ (or $\beta$) 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 the noise scale $\nu$.** We evaluate the accuracy of RNA models on perturbed MMLU and WMDP by varying $\nu$. As shown in Figure 3, we observed that increasing $\nu$ first leads to

improved accuracy of RNA models on perturbed MMLU while maintaining stable accuracy on WMDP. However, as $\nu$ continuous 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 5.2, 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 $\nu$ than PO.**

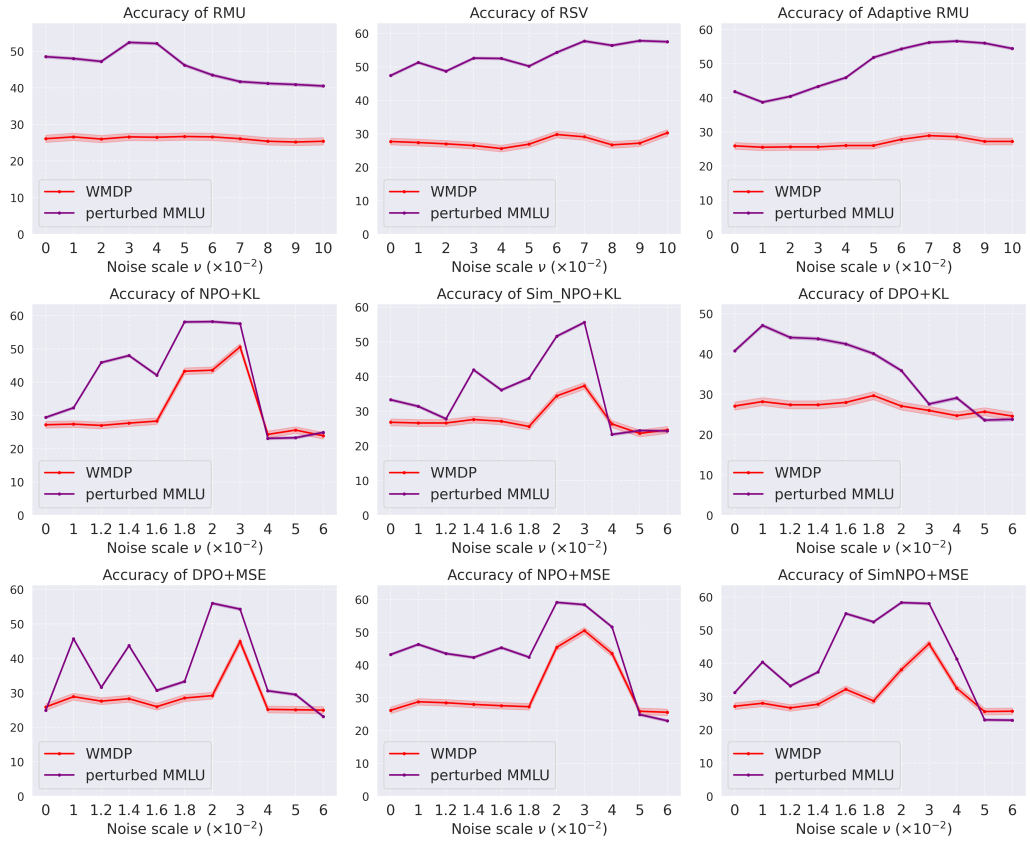


Figure 3: Accuracy of RNA models measured on perturbed MMLU Q&A and WMDP (avg. of Biology and Cyber) across different values of $\nu$.

**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 w.r.t these forget-tokens are synthesized following the procedure described in Section 6.1. As shown in Figure 4, 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.** See Appendix D for the evaluation of RNA models' robustness against $n$-gram similarity perturbations for $n \in \{4, 8, 16\}$.

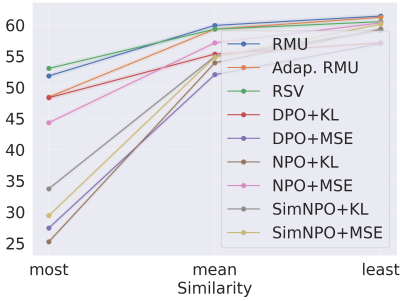


Figure 4: Accuracy of unlearned models on perturbed MMLU w.r.t bi-gram similarity perturbations.

## 7 Conclusion

This paper proposes RNA, a simple data augmentation 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.


## Acknowledgments

The authors thank Nguyen Minh Phuong for insightful comments and discussions.


## References


[1] 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*, pages 254–263. PMLR, 2018.

[2] 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.

[3] 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.

[4] 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.

[5] 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)*, pages 141–159. IEEE, 2021.

[6] Junyoung Byun, Hyojun Go, and Changick Kim. On the effectiveness of small input noise for defending against query-based black-box attacks. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3051–3060, 2022.

[7] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480, 2015. doi: 10.1109/SP.2015.35.

[8] 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*, pages 10732–10747, 2024.

[9] 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.

[10] A Feder Cooper, Christopher A Choquette-Choo, Miranda Bogen, Matthew Jagielski, Katja Filippova, Ken Ziyu Liu, Alexandra Chouldechova, Jamie Hayes, Yangsibo Huang, Niloofar Mireshghallah, et al. Machine unlearning doesn't do what you think: Lessons for generative ai policy, research, and practice. *arXiv preprint arXiv:2412.06966*, 2024.

[11] 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, pages 23733–23742, 2025.

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

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

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

[15] 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 *Neurips Safe Generative AI Workshop 2024*, 2024. URL https://openreview.net/forum?id=pVACX02m0p.

[16] 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. *arXiv preprint arXiv:2502.05374*, 2025.

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

[18] 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.

[19] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 588–597, 2019.

[20] 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.

[21] 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.

[22] 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.

[23] Nguyen Hung-Quang, Yingjie Lao, Tung Pham, Kok-Seng Wong, and Khoa D Doan. Understanding the robustness of randomized feature defense against query-based adversarial attacks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vZ6r9GMT1n.

[24] 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.

[25] 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.

[26] Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 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/.

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

[28] 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.

[29] Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan K Kummerfeld, and Rada Mihalcea. A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity. In *International Conference on Machine Learning*, pages 26361–26378. PMLR, 2024.

[30] 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, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 28525–28550. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/li24bc.html.

[31] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the european conference on computer vision (ECCV)*, pages 369–385, 2018.

[32] Michelle Lo, Fazl Barez, and Shay Cohen. Large language models relearn removed concepts. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 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/.

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

[34] 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.

[35] 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.

[36] 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.

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

[38] Aashiq Muhamed, Jacopo Bonato, Mona Diab, and Virginia Smith. Saes can improve unlearning: Dynamic sparse autoencoder guardrails for precision unlearning in llms. *arXiv preprint arXiv:2504.08192*, 2025.

[39] 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.

[40] 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. *arXiv preprint arXiv:2209.02299*, 2022.

[41] 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.

[42] Soumyadeep Pal, Changsheng Wang, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. Llm unlearning reveals a stronger-than-expected coreset effect in current benchmarks. *arXiv preprint arXiv:2504.10185*, 2025.

[43] 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.

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

[45] Zeyu Qin, Yanbo Fan, Hongyuan Zha, and Baoyuan Wu. Random noise defense against query-based black-box attacks. *Advances in Neural Information Processing Systems*, 34:7650–7663, 2021.

[46] 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.

[47] 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)*, pages 3982–3992, 2019.

[48] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in neural information processing systems*, 32, 2019.

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

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

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] Pratiksha Thaker, Shengyuan Hu, Neil Kale, Yash Maurya, Zhiwei Steven Wu, and Virginia Smith. Position: Llm unlearning benchmarks are weak measures of progress. *arXiv preprint arXiv:2410.02879*, 2024.

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

[59] 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.

[60] 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*, 2025. URL https://openreview.net/forum?id=6ESRicalFE.

[61] 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*, 2024. URL https://openreview.net/forum?id=ar8aRMrmod.

[62] 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, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 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/.

[63] 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.

[64] 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, pages 25769–25777, 2025.

[65] 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*, 2025. URL https://openreview.net/forum?id=Q1MHvGmhyT.

[66] 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.

[67] 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.

[68] 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.

# Appendix

## Table of Contents



## A Experimental Setup

### A.1 Datasets

**WMDP** [30] 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`.

**Wikitext** [37] is a language modeling dataset consisting of over $100$ milion tokens extracted from Wikipedia. Following [30], 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** [20] 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** [20] 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 from Maini et al. [36] as the preference answers for DPO+KL and DPO+MSE. For further details, we refer the reader to Appendix C of Maini et al. [36].

**Perturbed Query**

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 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.

### A.2 Prompt Templates

We use the lm-evaluation-harness framework [18] for evaluation. Each query is formulated as a default zero-shot QA prompt (Figure 6). Following the setting of prior work [57], 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. Additional experiments on RNA's performance with other forget-tokens can be found in Appendix D.

### A.3 Evaluation Metrics

Following [30], we 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:

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

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

**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 $66.67\%$.

### A.4 Implementation Details.

**Hyperparameters.** Models are fine-tuned using AdamW [33] 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 [30, 11], we update three layers of parameters $\{l, l-1, l-2\}$ of the model to save memory. 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 $c = 6.5$ 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 $c = 10$. For the original PO methods, we adopt the default hyperparameters used in prior work [65, 15]. 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 $(5, 10)$ 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 $\nu \in \{10^{-2}, 1.2\times10^{-2}, 1.4\times10^{-2}, 1.6\times10^{-2}, 1.8\times10^{-2}, 2\times10^{-2}, 3\times10^{-2}, 4\times10^{-2}, 5\times10^{-2}, 6\times10^{-2}, 7\times10^{-2}, 8\times10^{-2}, 9\times10^{-2}, 10^{-1}\}$ 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 included in the supplementary material. The perturbed MMLU QA datasets will be made publicly available.

# B Proofs

For clarity, we restate the theorems below.

## B.1 Proof of Theorem 4.2

**Theorem 4.2.** *If Assumption 4.1 holds, the change in the output representation of the generated token $x_i^r$ given the perturbed retain-query $x_{<i}^{r,\text{per}}$ and the benign retain-query $x_{<i}^r$ in the unlearned model $f^u$, defined as $\Delta = f^u(x_i^r|x_{<i}^{r,per}) - f^u(x_i^r|x_{<i}^r)$, follows the Normal distribution $\mathcal{N}(\mathbf{0}, \eta \boldsymbol{J}^\top \boldsymbol{J})$, where $\boldsymbol{J} = \nabla_{\mathbf{z}_{<i}^r} f^u(x_i^r|x_{<i}^r)$ is the Jacobian of $f^u(x_i^r|x_{<i}^r)$ with respect to $\mathbf{z}_{<i}^r$.*

*Proof.* Consider the output representation of the predicted token $x_i^r$ given the perturbed retain-query prefix $x_{<i}^{r,\text{per}}$ in the unlearned model $f^u(x_i^r|x_{<i}^{r,\text{per}})$. We show the claim by using the framework of the generative latent variable model (GLVM). Specifically, model $f^u$ generates token $x_i^r$ conditioned on a latent variable $\mathbf{z}_{<i}^{r,\text{per}}$ corresponding to the perturbed prefix $x_{<i}^{r,\text{per}}$, denoted as $f^u(x_i^r|\mathbf{z}_{<i}^{r,\text{per}})$. Under Assumption 4.1, the following holds:

$$f^u(x_i^r|\mathbf{z}_{<i}^{r,\text{per}}) = f^u(x_i^r|\mathbf{z}_{<i}^r + \boldsymbol{\epsilon}) \tag{16}$$

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

$$f^u(x_i^r|\mathbf{z}_{<i}^r + \boldsymbol{\epsilon}) \approx f^u(x_i^r|\mathbf{z}_{<i}^r) + \nabla_{\mathbf{z}_{<i}^r} f^u(x_i^r|\mathbf{z}_{<i}^r)^\top \boldsymbol{\epsilon} \tag{17}$$

$$f^u(x_i^r|\mathbf{z}_{<i}^r + \boldsymbol{\epsilon}) - f^u(x_i^r|\mathbf{z}_{<i}^r) \approx \nabla_{\mathbf{z}_{<i}^r} f^u(x_i^r|\mathbf{z}_{<i}^r)^\top \boldsymbol{\epsilon} \tag{18}$$

Let $\Delta = f^u(x_i^r|\mathbf{z}_{<i}^r + \boldsymbol{\epsilon}) - f^u(x_i^r|\mathbf{z}_{<i}^r)$, 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 $\boldsymbol{J} = \nabla_{\mathbf{z}_{<i}^r} f^u(x_i^r|\mathbf{z}_{<i}^r)$ is the Jacobian of $f^u(x_i^r|\mathbf{z}_{<i}^r)$ with respect to $\mathbf{z}_{<i}^r$. □

## B.2 Proof of Theorem 5.2

**Theorem 5.2.** *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 4.1 and Assumption 5.1 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 $\frac{1}{2} - \frac{1}{\pi} \arctan\left[\sqrt{\frac{\eta}{\nu}}\left(1 + \frac{||\boldsymbol{g}^{per}||_2}{||\boldsymbol{g}||_2}\right)^{-1}\right]$, where $\boldsymbol{g}^{per} = \nabla_{\mathbf{z}_{<i}^{r,per}} \mathcal{J}(f^u(x_i^r|x_{<i}^{r,per}))$ and $\boldsymbol{g} = \nabla_{\mathbf{z}_{<i}^r} \mathcal{J}(f^u(x_i^r|x_{<i}^r))$ are the gradients of the loss of generated token $x_i^r$ with respect to $\mathbf{z}_{<i}^{r,per}$ and $\mathbf{z}_{<i}^r$.*

*Proof.* Let us consider the generation of $x_i^r$ through the lens of a GLVM. The loss of $x_i^r$ given the latent representation $\mathbf{z}_{<i}^{r,\text{per}}$ of the prefix $x_{<i}^{r,\text{per}}$ in unlearned model $f^u$, is denoted by $f^u(x_i^r|\mathbf{z}_{<i}^{r,\text{per}})$. Under Assumption 4.1, the following holds:

$$\mathcal{J}(f^u(x_i^r|\mathbf{z}_{<i}^{r,\text{per}})) = \mathcal{J}(f^u(x_i^r|\mathbf{z}_{<i}^r + \boldsymbol{\epsilon})) \tag{19}$$

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

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

Rearranging Eqn. 20, we obtain the approximate change in loss:

$$\Delta\mathcal{J}^u \approx \nabla_{\mathbf{z}_{<i}^r}\mathcal{J}(f^u(x_i^r|\mathbf{z}_{<i}^r))^\top \boldsymbol{\epsilon} \tag{21}$$

Under Assumption 4.1 and Assumption 5.1, $\mathcal{J}(f^{\text{rna}}(x_i^r|\mathbf{z}_{<i}^{r,\text{per}}))$ and $\mathcal{J}(f^{\text{rna}}(x_i^r|\mathbf{z}_{<i}^r))$ can be expressed as:

$$\begin{aligned}
\mathcal{J}(f^{\text{rna}}(x_i^r|\mathbf{z}_{<i}^{r,\text{per}})) &= \mathcal{J}(f^{\text{rna}}(x_i^r|\mathbf{z}_{<i}^r + \boldsymbol{\epsilon})) && (22)\\
&\approx \mathcal{J}(f^u(x_i^r|\mathbf{z}_{<i}^r + \boldsymbol{\epsilon} + \boldsymbol{\delta}_1)) && (23)\\
&\approx \mathcal{J}(f^u(x_i^r|\mathbf{z}_{<i}^{r,\text{per}} + \boldsymbol{\delta}_1)) && (24)\\
&\approx \mathcal{J}(f^u(x_i^r|\mathbf{z}_{<i}^{r,\text{per}})) + \nabla_{\mathbf{z}_{<i}^{r,\text{per}}}\mathcal{J}(f^u(x_i^r|\mathbf{z}_{<i}^{r,\text{per}}))^\top \boldsymbol{\delta}_1 && (25)\\
\mathcal{J}(f^{\text{rna}}(x_i^r|\mathbf{z}_{<i}^r)) &\approx \mathcal{J}(f^u(x_i^r|\mathbf{z}_{<i}^r) + \boldsymbol{\delta}_2)) \\
&\approx \mathcal{J}(f^u(x_i^r|\mathbf{z}_{<i}^r)) + \nabla_{\mathbf{z}_{<i}^r}\mathcal{J}(f^u(x_i^r|\mathbf{z}_{<i}^r))^\top \boldsymbol{\delta}_2 && (26)
\end{aligned}$$

Substituting Eqn. 25 and Eqn. 26, the change in loss in RNA model $f^{\text{rna}}$ of predicted token $x_i^r$ is approximately:

$$\Delta\mathcal{J}^{\text{rna}} \approx \Delta\mathcal{J}^u + (\boldsymbol{g}^{\text{per}})^\top \boldsymbol{\delta}_1 - \boldsymbol{g}^\top \boldsymbol{\delta}_2, \tag{27}$$

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

The ratio of the RNA loss change to the original unlearned model loss change is:

$$\frac{\Delta\mathcal{J}^{\text{rna}}}{\Delta\mathcal{J}^u} \approx 1 + \frac{(\boldsymbol{g}^{\text{per}})^\top \boldsymbol{\delta}_1 - \boldsymbol{g}^\top \boldsymbol{\delta}_2}{\Delta\mathcal{J}^u} = 1 + \frac{(\boldsymbol{g}^{\text{per}})^\top \boldsymbol{\delta}_1 - \boldsymbol{g}^\top \boldsymbol{\delta}_2}{\boldsymbol{g}^\top \boldsymbol{\epsilon}} \tag{28}$$

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

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

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

$$\mathbb{P}\left[\frac{\Delta\mathcal{J}^{\text{rna}}}{\Delta\mathcal{J}^u} \leq 0\right] \approx \mathbb{P}\left[\frac{(\boldsymbol{g}^{\text{per}})^\top \boldsymbol{\delta}_1 - \boldsymbol{g}^\top \boldsymbol{\delta}_2}{\boldsymbol{g}^\top \boldsymbol{\epsilon}} \leq -1\right] \tag{29}$$

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 $\gamma = \sqrt{\frac{\nu}{\eta}}\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||_2}{||\boldsymbol{g}||_2}\right)$. The cumulative distribution function of Cauchy $\left(0, \sqrt{\frac{\nu}{\eta}}\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||_2}{||\boldsymbol{g}||_2}\right)\right)$ given by

$$F(x; x_0, \gamma) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x}{\sqrt{\frac{\nu}{\eta}}\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||_2}{||\boldsymbol{g}||_2}\right)}\right)$$

Thus, the probability is approximated:

$$\begin{aligned}
\mathbb{P}\left[\frac{(\boldsymbol{g}^{\text{per}})^\top \boldsymbol{\delta}_1 - \boldsymbol{g}^\top \boldsymbol{\delta}_2}{\boldsymbol{g}^\top \boldsymbol{\epsilon}} \leq -1\right] &= F(x = -1; x_0, \gamma) && (30)\\
&= \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{-1}{\sqrt{\frac{\nu}{\eta}}\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||_2}{||\boldsymbol{g}||_2}\right)}\right) && (31)\\
&= \frac{1}{2} - \frac{1}{\pi}\arctan\left[\sqrt{\frac{\eta}{\nu}}\left(1 + \frac{||\boldsymbol{g}^{\text{per}}||_2}{||\boldsymbol{g}||_2}\right)^{-1}\right] && (32)
\end{aligned}$$

□

## C Empirical Validation of Section 3

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

**Noise sensitivity of layers.** We formalize the forgetting through the lens of *noise sensitivity* [1]. Let $\mathbf{z}^f \in \mathbb{R}^{d_l}$ be the hidden states vector of forget-sample $x^f$ 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:

$$\mathcal{S}^g(\mathcal{D}_f) \stackrel{\text{def}}{=} \mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\mathbb{E}_{\mathbf{z}^f\sim\mathcal{D}_f} \frac{||\boldsymbol{J}_g(\mathbf{z}^f+\boldsymbol{\xi}) - \boldsymbol{J}_g(\mathbf{z}^f)||_2^2}{||\boldsymbol{J}_g(\mathbf{z}^f)||_2^2}, \tag{33}$$

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. 33. The max sequence length of each forget-sample is set to 512.

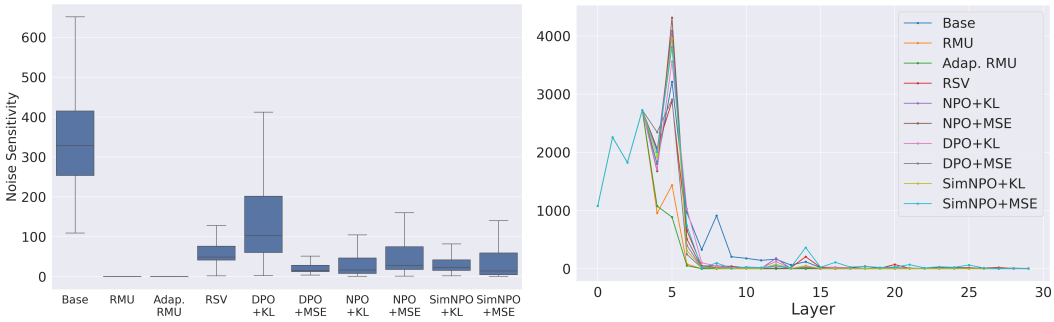


Figure 7: **Left**: noise sensitivity of layer $g = 8$ for the base model, PO models, and RM models. **Right**: Layer-wise noise sensitivity across all layers for the base model, PO models, and RM models.

**Results.** As shown in Figure 7 (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 7 (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. 33), 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 difficulty at the level of individual samples, sub-classes, or data subsets. We leave these promising directions for future work.

## D Robustness of RNA Against Hard Negative Forget-Tokens

In this section, we analyze RNA's robustness to forget-tokens that exhibit high $n$-gram similarity.

Table 1: Selected value of $\nu$ ($\times 10^{-2}$) 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 2: 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 3: 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** | 28.8 (−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 4: 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 5: 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) |

**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 [47], 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.

**Hyperparameters.** We utilize model checkpoints from the previous setting and perform evaluations accordingly. Results are reported for checkpoints selected based on the optimal noise scale $\nu$, as detailed in Table 1.

**Results.** RNA's performance is summarized in Table 2 through Table 5. 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.

## E 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.

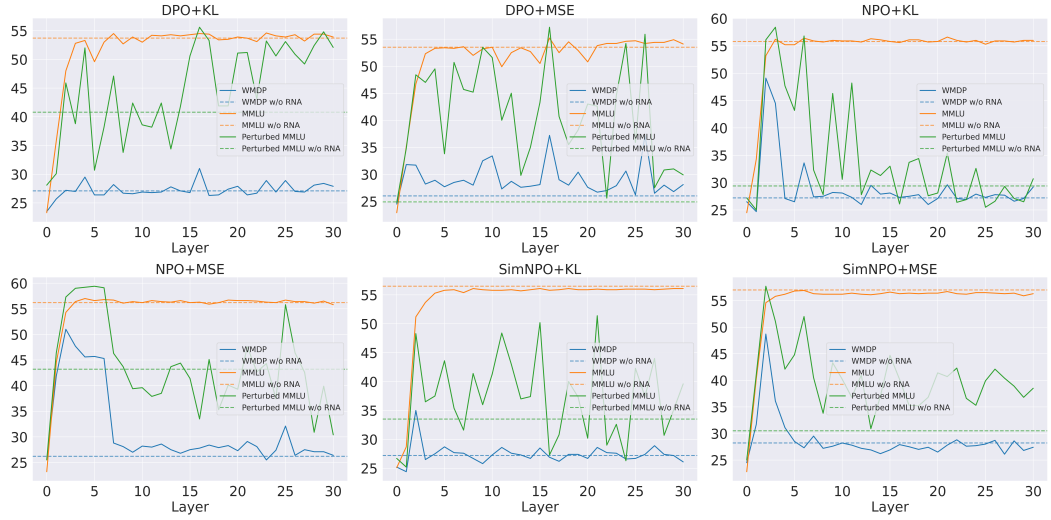


Figure 8: *Per-layer injection*: accuracy of RNA models on MMLU, perturbed MMLU and WMDP (avg. of Biology and Cyber) across different perturbed layers.

**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.** Figures 8–10 demonstrate that RNA generally improves the robustness of unlearned models. While Figures 8 and 9 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 10 further shows that *injecting noise into all layer is particularly effective at moderate noise levels* (*e.g.,* $1 \times 10^{-3}$). However, as the noise scale $\nu$ 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.

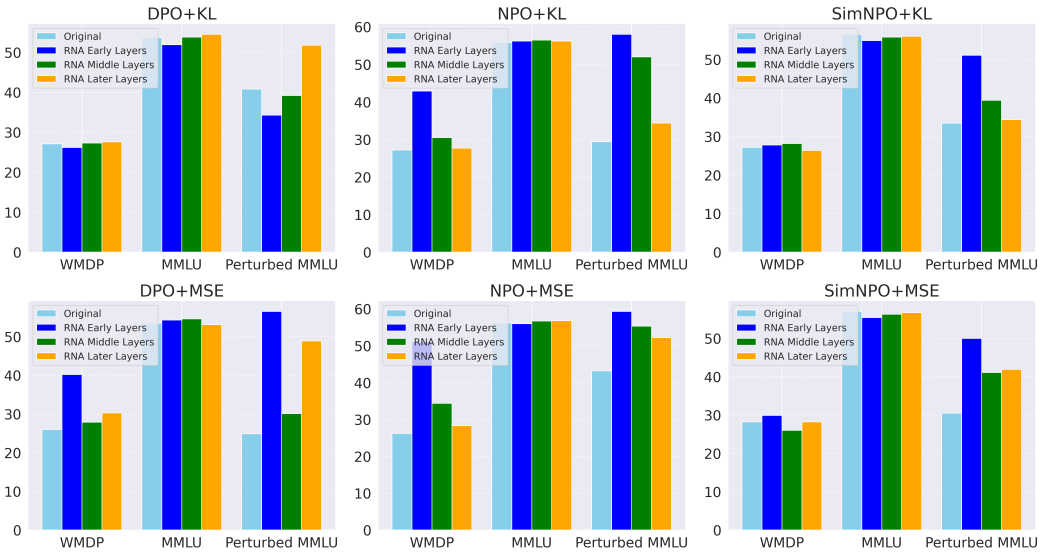


Figure 9: *Region-specific layer injection*: accuracy of RNA models on MMLU, perturbed MMLU and WMDP (avg. of Biology and Cyber) w.r.t early layers $(5, 6, 7)$, middle layers $(14, 15, 16)$, and later layers $(28, 29, 30)$.

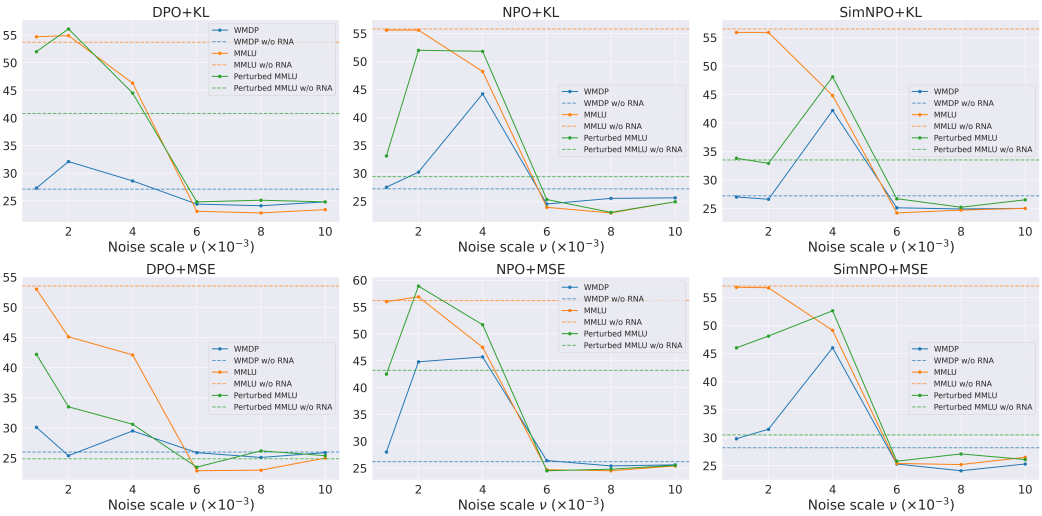


Figure 10: *Full-layer injection*: accuracy of RNA models on MMLU, perturbed MMLU and WMDP (avg. of Biology and Cyber).

## F Limitation and Future Work

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

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

**Future works.** RNA is designed for fine-tuning-based unlearning, which requires access to the model parameters. Future work exploring RNA as an inference-time intervention method could be

promising. One potential direction is to conduct analysis on the optimal coefficient or noise scale to further improve RNA's effectiveness.

## G Broader Impact

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 providing a valuable solution for developing more secure and reliable machine unlearning systems.