On Effects of Steering Latent Representation for Large Language Model Unlearning

Dang Huu-Tien[1], **Tin Pham**[1], **Hoang Thanh-Tung**[2], **and Naoya Inoue**[1,3]

[1]Japan Advanced Institute of Science and Technology
[2]VNU University of Engineering and Technology, Vietnam
[3]RIKEN
[1]{s2310417, tinpham, naoya-i}@jaist.ac.jp, [2]htt210@gmail.com

Abstract

Representation Misdirection for Unlearning (RMU), which steers model representation in the intermediate layer to a target random representation, is an effective method for large language model (LLM) unlearning. Despite its high performance, the underlying cause and explanation remain underexplored. In this paper, we theoretically demonstrate that steering forget representations in the intermediate layer reduces token confidence, causing LLMs to generate wrong or nonsense responses. We investigate how the coefficient influences the alignment of forget-sample representations with the random direction and hint at the optimal coefficient values for effective unlearning across different network layers. We show that RMU unlearned models are robust against adversarial jailbreak attacks. Furthermore, our empirical analysis shows that RMU is less effective when applied to the middle and later layers in LLMs. To resolve this drawback, we propose *Adaptive RMU*—a simple yet effective alternative method that makes unlearning effective with most layers. Extensive experiments demonstrate that Adaptive RMU significantly improves the unlearning performance compared to prior art while incurring no additional computational cost.

1 Introduction

LLMs achieved remarkable performance through pretraining on large amounts of internet texts and rigorous alignment processes for safety enhancement. Despite the immense effort in safety research, LLMs are still vulnerable to adversarial jailbreak attacks and can exhibit unwanted behaviors (Shah et al. 2023; Zou et al. 2023b; Jones et al. 2023; Yuan et al. 2024; Wei, Haghtalab, and Steinhardt 2024).

Machine Unlearning (Cao and Yang 2015; Bourtoule et al. 2021; Nguyen et al. 2022; Xu et al. 2023; Liu et al. 2024c) has emerged as a promising method for mitigating unforeseen risks in LLMs before deployment. Li et al. (2024b) introduced Representation Misdirection for Unlearning (RMU)—an unlearning method that steers the representations of forget-samples (*i.e.* samples that the model should forget) toward a random representation while keeping the representations of retain-samples (*i.e.* samples that the model should remember) unchanged. RMU significantly degrades models' accuracy on forget-tasks, while only slightly affecting the performance on retain-tasks and demonstrates stronger robustness against adversarial jailbreak attacks. However, the reason for RMU's effectiveness is not well understood, hindering the development of better unlearning algorithms. In this paper, we make the following contributions:

- We theoretically analyze the impact of the RMU method on LLM unlearning.
- We investigate the connection between RMU and adversarial robustness. We demonstrate that RMU impedes the adversary's ability to determine optimal updates for generating adversarial samples, thus improving the adversarial robustness of the unlearned model.
- We empirically show that the RMU forget loss, which minimizes the mean squared error (MSE) between forget representation and a fixed scaled random vector, fails to converge when the norm of the forget representation is larger than the scaling coefficient, making RMU less effective when applied to middle and last layers in LLMs.
- To overcome RMU's limitation, we introduce *Adaptive RMU*—a variant that adaptively adjusts the coefficient value based on the norm of the forget representation. Experimental results show that Adaptive RMU achieves higher drop-in-accuracy for forget knowledge, maintaining high performance on general knowledge, and enables effective unlearning for most layers without incurring additional computational overhead.

2 Background and Related Work

Machine Unlearning. A natural unlearning approach is leave-some-out retraining: retraining the model from scratch without the forget samples. However, this method becomes more computationally expensive as the size of datasets and modern deep networks grows. Existing works focus on approximating unlearning (Warnecke et al. 2021; Izzo et al. 2021; Sekhari et al. 2021; Isonuma and Titov 2024) using influence function (Koh and Liang 2017; Grosse et al. 2023), gradient ascent (Thudi et al. 2022), second-order approximation (Jia et al. 2024), negative preference optimization (Zhang et al. 2024b), and embedding corrupted (Liu et al. 2024a). Other views on the landscape of machine unlearning include: unlearning in text classification (Ma et al. 2022), image classification and recognition (Ginart et al.

2019; Golatkar, Achille, and Soatto 2020; Fan et al. 2024; Choi and Na 2023; Cha et al. 2024), image-to-image generative models (Li et al. 2024a), diffusion models (Gandikota et al. 2023; Zhang et al. 2024a; Kumari et al. 2023; Bui et al. 2024), multimodal unlearning (Cheng and Amiri 2023), federated unlearning (Romandini et al. 2024; Wang et al. 2022; Che et al. 2023; Halimi et al. 2022; Jeong, Ma, and Houmansadr 2024), graph unlearning (Chen et al. 2022; Chien, Pan, and Milenkovic 2023; Wu et al. 2023a; Cheng et al. 2023; Dukler et al. 2023; Zhu, Li, and Hu 2023; Li et al. 2024c; Tan et al. 2024), recommender systems (Zhang et al. 2023; Chen et al. 2024; Li et al. 2023; Wang et al. 2025), certified minimax unlearning (Liu et al. 2024b), targeted types of unlearning information (Cooper et al. 2024), and evaluation on unlearning (Lynch et al. 2024; Hayes et al. 2024; Shi et al. 2024a,b).

LLM Unlearning. Due to the large size of the parameters and training data, LLM poses a new challenge to unlearning. Recent studies in LLM unlearning mainly focus on task or context-specific settings such as unlearning copyrighted material from the Harry Potter series (Eldan and Russinovich 2023), in-context unlearning (Pawelczyk, Neel, and Lakkaraju 2024), fictitious unlearning (Maini et al. 2024), specific harmful input-output (Yao, Xu, and Liu 2023; Liu et al. 2024d), sensitive and private information (Jang et al. 2023; Wu et al. 2023b; Patil, Hase, and Bansal 2024), gender bias (Belrose et al. 2023) or concepts (Hong et al. 2024; Bui et al. 2024). More recently, Li et al. (2024b) consider unlearning an entire distribution of hazardous knowledge given limited samples.

Notation & problem formulation. Let $\mathcal{D}_{\text{forget}}$ and $\mathcal{D}_{\text{retain}}$ be the forget and retain sets, respectively. Let $f_\theta : \mathbb{R}^{n\times d} \mapsto \mathbb{R}^{n\times|V|}$ be an autoregressive LLM parameterized by θ that maps a prompt input $x_{1:n}$ consisting of n tokens $\{x_1, x_2, ..., x_n\}$ to an output of probability distributions over the vocabulary V. We denote $h_\theta^{(l)}(x)$ the *averaged* hidden states of all tokens in $x_{1:n}$ obtained from the l-th layer of f_θ. For simplicity, throughout this paper, we use $h^{(l)}(x)$ to present $h_\theta^{(l)}(x)$. For operators, we denote $\circ$ as the decomposition operator, and $||\cdot||$ is the Euclidean norm. Our goal is to unlearn the undesired harmful knowledge $\mathcal{D}_{\text{forget}}$ from f_θ while retaining general knowledge $\mathcal{D}_{\text{retain}}$. Unlearned models should be robust to knowledge recovery attacks that attempt to recover harmful knowledge from the model.

Representation Misdirection for Unlearning (RMU; Li et al. (2024b)) is a fine-tuning based unlearning method inspired by representation engineering (Zou et al. 2023a) that steers the model's representation of forget samples $x_F \in \mathcal{D}_{\text{forget}}$ to a random vector and regularizes the model representation of retain samples $x_R \in \mathcal{D}_{\text{retain}}$ back to the original model representation, by optimizing the MSE loss:

$$\mathcal{L} = \mathbb{E}_{x_F\in\mathcal{D}_{\text{forget}}}||h^{(l)}_{\theta^{\text{unlearn}}}(x_F) - c\boldsymbol{u}||_2^2 + \alpha\mathbb{E}_{x_R\in\mathcal{D}_{\text{retain}}}||h^{(l)}_{\theta^{\text{unlearn}}}(x_R) - h^{(l)}_{\theta^{\text{frozen}}}(x_R)||_2^2, \quad (1)$$

where θ^{unlearn} and θ^{frozen} are parameters of the update model and frozen model respectively, $\boldsymbol{u}$ is a fixed random unit vector where each element is sampled from Uniform distribution $U(0,1)$, $c \in \mathbb{R}$ is a fixed scaling coefficient and $\alpha \in \mathbb{R}$ is a retain weight. RMU updates θ^{unlearn} toward the direction of the gradient of the loss $\mathcal{L}$ using gradient descent.

3 Theoretical Analysis

3.1 The Confidence of Tokens Generated by RMU Models

In general, samples from the shifted distribution (such as wrong label or out-of-distribution) are associated with smaller "confidence" scores such as softmax probability (Hendrycks and Gimpel 2017; Northcutt, Jiang, and Chuang 2021), maximum logit (Hendrycks et al. 2022; Wei et al. 2022), ℓ^2-distance (Sun et al. 2022), energy score (Liu et al. 2020), and cosine similarity (Ngoc-Hieu et al. 2023). Recently, LLM has shown a tendency to produce a lower (higher) confidence in its incorrect (correct) answers in multiple-choice Q&A (Plaut, Nguyen, and Trinh 2024). Building on previous works, we hypothesized that the *logit* of generated tokens by RMU models exhibit randomness. As seen by a deep network, such randomization signifies low confidence in the logit, resulting in nonsensical or incorrect responses. To validate the hypothesis, we conducted an analysis of the logits of generated tokens produced by RMU models. To facilitate subsequent analysis, we make the following definition and assumption.

Definition 1. *(Unlearned model & logit of forget-tokens on unlearned model). Let $f^{(l:k)} = g^{(l:k)} \circ h^{(l)}$, where $g^{(l:k)}$ be the transformation from layer l to layer k of network f, for any two layers $k > l$; $l \in [1...L]$. We define the unlearned model $f^{\text{unlearn}} = \boldsymbol{W}(f^{(l:L),\text{steered}}) = \boldsymbol{W}(g^{(l:L)} \circ h^{(l),\text{steered}})$, $h^{(l),\text{steered}}$ is the steered representation of the given input at layer l and $\boldsymbol{W}$ is the unembedding matrix which maps output hidden states back to the vocabulary space. Given a forget input $x_{F,1:n}$, the logit of the next token $x_{F,n+1}$ obtained from unlearned model $f^{unlearn}$ is defined as:*

$$\begin{aligned} f^{unlearn}(x_{F,n+1}|x_{F,1:n}) &= \boldsymbol{W}f^{(l:L),steered}(x_{F,n+1}|x_{F,1:n}) \\ &= \boldsymbol{W}(g^{(l:L)} \circ h^{(l),\text{steered}})(x_{F,n+1}|x_{F,1:n}) \\ &= \boldsymbol{W}g^{(l:L)}(h^{(l),\text{steered}}(x_{F,n+1}|x_{F,1:n})) \end{aligned} \quad (2)$$

Assumption 1. *A well-unlearned model shifts the representation of all tokens in a forget-sample $x_{F,1:n}$ at layer l to a scaled random vector $c\boldsymbol{u}$. More concretely,*

$$h^{(l),\text{steered}}(x_{F,i}) = c\boldsymbol{u} + \boldsymbol{\epsilon}, \quad (3)$$

where $x_{F,i}$ is the i-th token in x_F, $\boldsymbol{\epsilon}$ is a small error. Without losing generality, we assume that $\boldsymbol{\epsilon}$ is sampled from Normal distribution $\mathcal{N}(\boldsymbol{0}, \eta\boldsymbol{I})$, where $\eta\boldsymbol{I}$ is the covariance matrix, $\eta \in \mathbb{R}$.

Proposition 1. *If Assumption 1 holds, by Definition 1, the logit value of forget token $x_{F,n+1}$ generated by unlearned model f^{unlearn} given as $f^{\text{unlearn}}(x_{F,n+1}|x_{F,1:n})$ follows the Normal distribution $\mathcal{N}\left(\boldsymbol{W}g^{(l:L)}(\boldsymbol{z}), \eta\boldsymbol{W}\nabla_{\boldsymbol{z}}g^{(l:L)}(\boldsymbol{z})^\top\nabla_{\boldsymbol{z}}g^{(l:L)}(\boldsymbol{z})\boldsymbol{W}^\top\right)$, where $\boldsymbol{z} = c\boldsymbol{u}$.*

Proof. Assumption 1 implies that in a well-unlearned model, token $x_{F,n+1}$ is independent of the previous tokens, thus we have:

$$h^{(l),\text{steered}}(x_{n+1}|x_{F,1:n}) \approx h^{(l),\text{steered}}(x_{F,n+1}) = c\boldsymbol{u} + \boldsymbol{\epsilon} \tag{4}$$

Denote $\boldsymbol{z} = c\boldsymbol{u}$. Substituting Eqn. 4 into Eqn. 2, we get:

$$f^{\text{unlearn}}(x_{F,n+1}|x_{F,1:n}) \approx \boldsymbol{W} g^{(l:L)}(\boldsymbol{z} + \boldsymbol{\epsilon}) \tag{5}$$

Since $\boldsymbol{\epsilon}$ is small, we approximate the function $g^{(l:L)}(\boldsymbol{z} + \boldsymbol{\epsilon})$ by its first-order derivative:

$$f^{\text{unlearn}}(x_{F,n+1}|x_{F1:n}) \approx \boldsymbol{W}(g^{(l:L)}(\boldsymbol{z}) + \nabla_{\boldsymbol{z}} g^{(l:L)}(\boldsymbol{z})^\top \boldsymbol{\epsilon}) \tag{6}$$

Given that $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \eta \boldsymbol{I})$, by applying the affine transformation property of the multivariate normal distribution, we get:

$$\begin{aligned} &f^{\text{unlearn}}(x_{F,n+1}|x_{F,1:n}) \\ &\sim \mathcal{N}\left(\boldsymbol{W} g^{(l:L)}(\boldsymbol{z}), \eta \boldsymbol{W} \nabla_z g^{(l:L)}(\boldsymbol{z})^\top \nabla_z g^{(l:L)}(\boldsymbol{z}) \boldsymbol{W}^\top\right) \end{aligned} \tag{7}$$

Since $\boldsymbol{u} \sim U(0,1)$, then $\boldsymbol{z} \sim U(0,c)$. By definition of variance, we have: $\text{Var}(\boldsymbol{z}) = \text{Var}(c\boldsymbol{u}) = c^2\text{Var}(\boldsymbol{u})$. □

Proposition 1 suggests that the variance of $f^{\text{unlearn}}(x_{F,n+1}|x_{F,1:n})$ is controlled by (i) η: a scalar variance and (ii) $\boldsymbol{W}\nabla_{\boldsymbol{z}} g^{(l:L)}(\boldsymbol{z})^\top \nabla_{\boldsymbol{z}} g^{(l:L)}(\boldsymbol{z}) \boldsymbol{W}^\top$: the product of $\boldsymbol{W}\nabla_{\boldsymbol{z}} g^{(L)}(\boldsymbol{z})^\top$ and $\nabla_{\boldsymbol{z}} g^{(L)}(\boldsymbol{z})\boldsymbol{W}^\top$. If $f^{\text{unlearn}}(x_{F,n+1}|x_{F,1:n})$ has high variance, the logit values are more random. Since $\boldsymbol{\epsilon}$ presents a small error, then $\boldsymbol{\epsilon}$ varies for different inputs x_F. This variation makes it difficult to control the variance of the logit by η. The main effect depend on $\boldsymbol{W}\nabla_{\boldsymbol{z}} g^{(l:L)}(\boldsymbol{z})^\top \nabla_{\boldsymbol{z}} g^{(l:L)}(\boldsymbol{z}) \boldsymbol{W}^\top$. While the unembedding matrix $\boldsymbol{W}$ is unchanged after unlearning, the product $\nabla_{\boldsymbol{z}} g^{(l:L)}(\boldsymbol{z})^\top \nabla_{\boldsymbol{z}} g^{(l:L)}(\boldsymbol{z})$ varies depending on the specific characteristics of sub-networks $g^{(l:L)}$ and input $\boldsymbol{z} = c\boldsymbol{u}$. Unfortunately, $g^{(l:L)}$ is a composition of transformer layers, which is highly nonlinear, making it difficult to have a complete analysis. The variance of $\boldsymbol{z}$, derived as $\text{Var}(\boldsymbol{z}) = c^2\text{Var}(\boldsymbol{u})$, is proportional to c; *i.e.* when c gets larger, the variance of $\boldsymbol{z}$ is higher. This could increase the variability of $g^{(l:L)}(\boldsymbol{z})$ and the gradient $\nabla_{\boldsymbol{z}} g^{(l:L)}(\boldsymbol{z})$. *A larger c could introduces more randomness to the logit*. We conduct an empirical analysis to understand the confidence of generated tokens by RMU models in Section 4.1.

3.2 The Effect of Coefficient c on Forget-sample Representations

RMU forget loss steers forget-sample representation $h^{(l)}(x_F)$ aligns with a random direction given by $\boldsymbol{u}$ and scales the magnitude of $h^{(l)}(x_F)$ to c (Eqn 1). While vector $\boldsymbol{u}$ is predetermined before unlearning, the magnitude of $h^{(l)}(x_F)$ varies depending on input x_F and specific properties of layer l. This raises the following research questions:
RQ1 (Direction): *"How does the coefficient c influence the alignment between $h^{(l)}(x_F)$ with $\boldsymbol{u}$."*
RQ2 (Magnitude): *"What is the optimal value of the coefficient c for effectively unlearning with different layers."*

Unlearning as minimizing the noise sensitivity. We aim to answer these questions by analyzing the unlearning problem under a noise compression view. We consider the output of a transformation $f^{(l:k)}$ on input x: $f^{(l:k)}(x) = (g^{(l:k)} \circ h^{(l)})(x) = g^{(l:k)}\left(h^{(l)}(x)\right)$. Suppose we compress a noise vector $\boldsymbol{\xi}$ to the representation $h^{(l)}$ of layer l at input x, then the output become $g^{(l:k)}\left(h^{(l)}(x) + \boldsymbol{\xi}\right)$. Naturally, if layer $g^{(l:k)}$ is robust (less sensitive) to noise $\boldsymbol{\xi}$, then $\boldsymbol{\xi}$ has a small effect on the output of $g^{(l:k)}$ *i.e.* the normalized squared norm

$$\Phi(g^{(l:k)}, x) = \frac{||g^{(l:k)}\left(h^{(l)}(x) + \boldsymbol{\xi}\right) - g^{(l:k)}\left(h^{(l)}(x)\right)||^2}{||g^{(l:k)}\left(h^{(l)}(x)\right)||^2} \tag{8}$$

is small. In contrast, a higher $\Phi(g^{(l:k)}, x)$ mean $g^{(l:k)}$ is higher sensitive to noise $\boldsymbol{\xi}$ at input x. For a dataset $\mathcal{D}_{\text{forget}}$, we define the *noise sensitivity* of a layer $g^{(l:k)}$ w.r.t $\boldsymbol{\xi}$ on $\mathcal{D}_{\text{forget}}$ as:

$$\begin{aligned} &\Phi(g^{(l:k)}, \mathcal{D}_{\text{forget}}) \\ &= \frac{||g^{(l:k)}(\hat{h}^{(l)}(x_F) + \boldsymbol{\xi}) - g^{(l:k)}(\hat{h}^{(l)}(x_F))||^2}{||g^{(l:k)}(\hat{h}^{(l)}(x_F))||^2}, \end{aligned} \tag{9}$$

where $\hat{h}^{(l)}(x_F)$ is the mean of $h^{(l)}(x_F)$ over $x_F \in \mathcal{D}_{\text{forget}}$. During unlearning, RMU steers $h^{(l)}(x_F)$ for all $x_F \in \mathcal{D}_{\text{forget}}$ to the fixed vector $c\boldsymbol{u} + \boldsymbol{\epsilon}$ *i.e.* $||g^{(l:k)}(c\boldsymbol{u} + \boldsymbol{\epsilon}) - g^{(l:k)}(\hat{h}^{(l)}(x_F))||^2$ is minimized. If we let $\boldsymbol{\xi} = c\boldsymbol{u} + \boldsymbol{\epsilon} - \hat{h}^{(l)}(x_F)$, we can define the unlearning problem as minimizing the noise sensitivity of the layer. This objective is described by

$$\min \frac{||g^{(l:k)}(c\boldsymbol{u} + \boldsymbol{\epsilon}) - g^{(l:k)}(\hat{h}^{(l)}(x_F))||^2}{||g^{(l:k)}(\hat{h}^{(l)}(x_F))||^2} \tag{10}$$

While $g^{(l:k)}$ is a composition of transformer layers, which is hard to expand it in term of c. Therefore, we propose to use the Jacobian matrix $\boldsymbol{J}^{(l:k)}(x_F)$—a linearized of $g^{(l:k)}$ at x_F—which describes the change in the output of $g^{(l:k)}$ due to a noise perturbed in the input $\hat{h}^{(l)}(x_F)$. For simplification, we write $\hat{h}^{(l)}$, $\boldsymbol{J}^{(l:k)}$ instead of $\hat{h}^{(l)}(x_F)$, $\boldsymbol{J}^{(l:k)}(x_F)$ respectively. The objective becomes

$$\min \frac{||\boldsymbol{J}^{(l:k)}(c\boldsymbol{u} + \boldsymbol{\epsilon}) - \boldsymbol{J}^{(l:k)}\hat{h}^{(l)}||^2}{||\boldsymbol{J}^{(l:k)}\hat{h}^{(l)}||^2} \tag{11}$$

Since $\boldsymbol{J}^{(l:k)}$ is a linear transformation, then

$$||\boldsymbol{J}^{(l:k)}(c\boldsymbol{u} + \boldsymbol{\epsilon}) - \boldsymbol{J}^{(l:k)}\hat{h}^{(l)}||^2 = ||\boldsymbol{J}^{(l:k)}(c\boldsymbol{u} + \boldsymbol{\epsilon} - \hat{h}^{(l)})||^2 \tag{12}$$

Let $\boldsymbol{v} = \boldsymbol{\epsilon} - \hat{h}^{(l)}$. By definition of the squared norm, we have:

$$\begin{aligned} ||\boldsymbol{J}^{(l:k)}(c\boldsymbol{u} + \boldsymbol{v})||^2 &= (\boldsymbol{J}^{(l:k)}(c\boldsymbol{u} + \boldsymbol{v}))^\top \boldsymbol{J}^{(l:k)}(c\boldsymbol{u} + \boldsymbol{v}) \\ &= (c\boldsymbol{u} + \boldsymbol{v})^\top \boldsymbol{J}^{(l:k)\top} \boldsymbol{J}^{(l:k)}(c\boldsymbol{u} + \boldsymbol{v}) \end{aligned} \tag{13}$$

Let matrix $\boldsymbol{A} = \boldsymbol{J}^{(l:k)\top}\boldsymbol{J}^{(l:k)}$. Expand the right-hand side of Eqn. 13, we get:

$$||\boldsymbol{J}^{(l:k)}(c\boldsymbol{u}+\boldsymbol{v})||^2 = (c\boldsymbol{u})^\top \boldsymbol{A}c\boldsymbol{u} + (c\boldsymbol{u})^\top \boldsymbol{A}\boldsymbol{v} + \boldsymbol{v}^\top \boldsymbol{A}c\boldsymbol{u} + \boldsymbol{v}^\top \boldsymbol{A}\boldsymbol{v} \quad (14)$$

Since $\boldsymbol{A}$ is a symmetric matrix (*i.e.* $\boldsymbol{A}^\top = \boldsymbol{A}$), then

$$(c\boldsymbol{u})^\top \boldsymbol{A}\boldsymbol{v} = (c\boldsymbol{u})^\top \boldsymbol{A}^\top \boldsymbol{v} = (\boldsymbol{A}c\boldsymbol{u})^\top \boldsymbol{v} = \boldsymbol{v}^\top \boldsymbol{A}c\boldsymbol{u} \quad (15)$$

Substituting $(c\boldsymbol{u})^\top \boldsymbol{A}\boldsymbol{v} = \boldsymbol{v}^\top \boldsymbol{A}c\boldsymbol{u}$ into Eqn. 14 we get:

$$||\boldsymbol{J}^{(l:k)}(c\boldsymbol{u}+\boldsymbol{v})||^2 = c^2\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{u} + 2c\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{v} + \boldsymbol{v}^\top \boldsymbol{A}\boldsymbol{v} \quad (16)$$

Substituting Eqn. 16 into Eqn. 11, the objective becomes

$$\min \frac{c^2\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{u} + 2c\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{v} + \boldsymbol{v}^\top \boldsymbol{A}\boldsymbol{v}}{||\boldsymbol{J}^{(l:k)}\hat{h}^{(l)}||^2} \quad (17)$$

Taking its derivative w.r.t c and set it to zero:

$$\frac{2\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{u}c + 2\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{v}}{||\boldsymbol{J}^{(l:k)}\hat{h}^{(l)}||^2} = 0 \quad (18)$$

Since $||\boldsymbol{J}^{(l:k)}\hat{h}^{(l)}||^2$ is not zero, solve for c:

$$\begin{aligned} c &= -\frac{\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{v}}{\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{u}} = \frac{\boldsymbol{u}^\top \boldsymbol{J}^{(l:k)\top}\boldsymbol{J}^{(l:k)}(\hat{h}^{(l)} - \boldsymbol{\epsilon})}{\boldsymbol{u}^\top \boldsymbol{J}^{(l:k)\top}\boldsymbol{J}^{(l:k)}\boldsymbol{u}} \\ &= \frac{(\boldsymbol{J}^{(l:k)}\boldsymbol{u})^\top \boldsymbol{J}^{(l:k)}(\hat{h}^{(l)} - \boldsymbol{\epsilon})}{||\boldsymbol{J}^{(l:k)}\boldsymbol{u}||^2} \\ &= \frac{||\boldsymbol{J}^{(l:k)}(\hat{h}^{(l)} - \boldsymbol{\epsilon}))||}{||\boldsymbol{J}^{(l:k)}\boldsymbol{u}||} \cos\Big(\boldsymbol{J}^{(l:k)}\boldsymbol{u}, \boldsymbol{J}^{(l:k)}(\hat{h}^{(l)} - \boldsymbol{\epsilon})\Big) \end{aligned} \quad (19)$$

Since $\frac{||\boldsymbol{J}^{(l:k)}(\hat{h}^{(l)} - \boldsymbol{\epsilon})||}{||\boldsymbol{J}^{(l:k)}\boldsymbol{u}||}$ is positive, then c *and* $\cos\Big(\boldsymbol{J}^{(l:k)}\boldsymbol{u}, \boldsymbol{J}^{(l:k)}(\hat{h}^{(l)} - \boldsymbol{\epsilon})\Big)$ *are positively correlated.*

This means smaller (larger) c indicates less (more) *alignment* between $\boldsymbol{J}^{(l:k)}\boldsymbol{u}$ and $\boldsymbol{J}^{(l:k)}(\hat{h}^{(l)} - \boldsymbol{\epsilon})$. Given that the Jacobian $\boldsymbol{J}^{(l:k)}$ describes how small changes in the input lead to changes in the output using linear approximation around a given point. If $\boldsymbol{J}^{(l:k)}$ does not vary drastically, it will not significantly alter the directions of $\boldsymbol{u}$ and $\hat{h}^{(l)} - \boldsymbol{\epsilon}$. In such cases, $\boldsymbol{J}^{(l:k)}$ will have a small effect on directional alignment, preserving the relative angles between $\boldsymbol{u}$ and $\hat{h}^{(l)} - \boldsymbol{\epsilon}$. Here, reasonably, $\boldsymbol{u}$ *and* $\hat{h}^{(l)}$ *are becoming more aligned as* c *increases* since error $\boldsymbol{\epsilon} \to \mathbf{0}$ as unlearning becomes more accurate.

The above discussion does not directly address RQ2. However, the definition of the noise sensitivity suggests that the noise sensitivity of layer $g^{(l:k)}$ is characterized by the inherent properties of $g^{(l:k)}$, the representation $\hat{h}^{(l)}(x_F)$ (which is fixed) and the perturbed noise $\boldsymbol{\xi}$. If $\boldsymbol{\xi}$ is predetermined, the noise sensitivity of $g^{(l:k)}$ depends solely on its properties. This suggest the following experiment: we compute $\hat{h}^{(l)}(x_F)$—the mean of $h^{(l)}(x_F)$ over a set of input $x_F \in \mathcal{D}_{\text{forget}}$, compress a fix perturbed noise $\boldsymbol{\xi}$ into $\hat{h}^{(l)}(x_F)$.

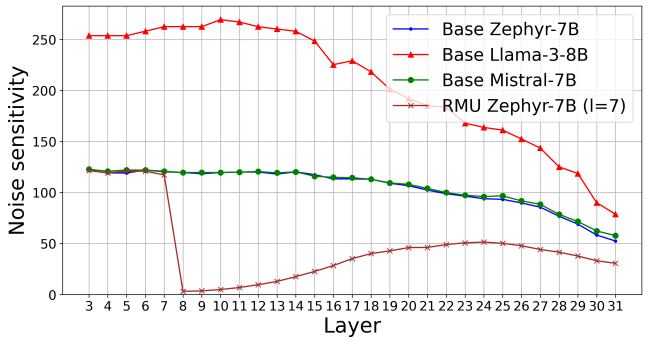


Figure 1: Noise sensitivity of layer $g^{(l:k)}$, for $k \in [3...31]$ in base Zephyr-7B, base Llama-3-8B, base Mistral-7B, and RMU Zephyr-7B model. In the base models, a deeper layer has lower noise sensitivity, while the noise sensitivity is minimized in the RMU model (compress noise into $h^{(7)}$, the noise sensitivity of layer $k = 8$ is minimized).

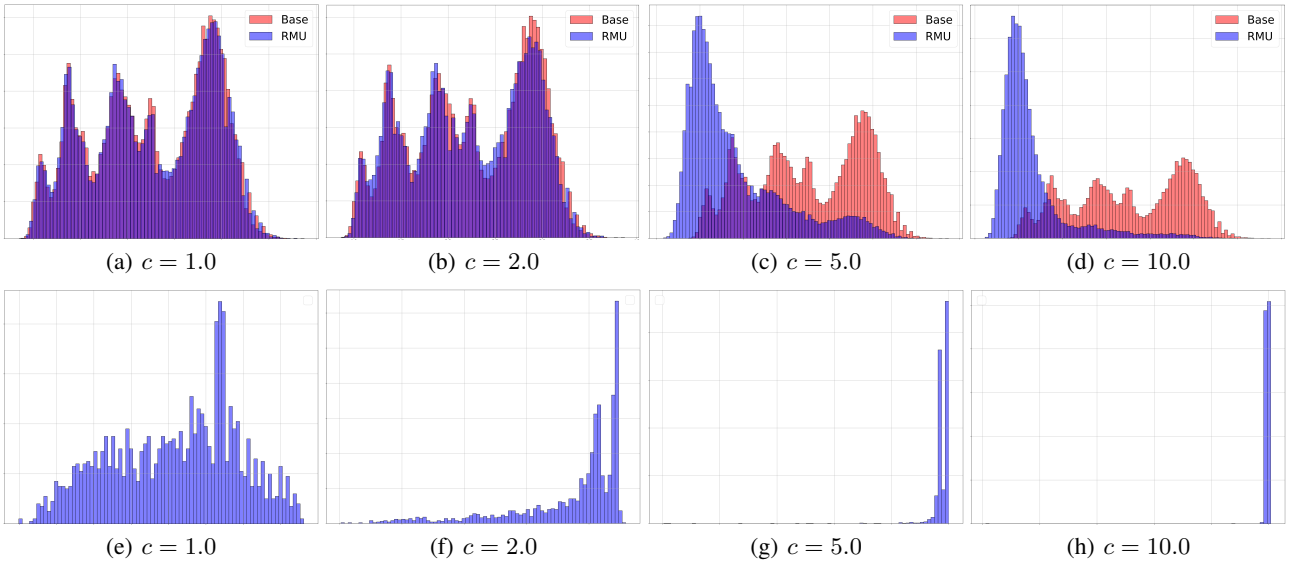
We then calculate the noise sensitivity of $g^{(l:k)}$ for different layers. Fig. 1 shows the noise sensitivity of layers across different models. We empirically observed that: *the noise sensitivity decreases as layers go deeper and vary across different models*. Since noise sensitivity describes a layer's robustness to noise, higher noise sensitivity means $g^{(l:k)}$ requires smaller noise to produce the same level of output randomness, while lower noise sensitivity means it requires larger noise. In other words, early layers require smaller noise $\boldsymbol{\xi}$ (smaller c) whereas later layers require larger noise $\boldsymbol{\xi}$ (larger c). We present an empirical experiment to verify our analysis in Section. 4.3.

3.3 Robustness of RMU Models to Adversarial Jailbreak Attacks

RMU algorithms not only erase undesirable knowledge (forgotten knowledge) but also prevent it from resurfacing when attacked by adversarial jailbreak attacks. Li et al. (2024b) present a qualitvative experiment by employing Greedy Coordinate Gradient (GCG; Zou et al. (2023b)) to jailbreak RMU models to extract the answer given a harmful input. As a result, RMU models demonstrate strong robustness against GCG attack. We conduct an analysis from the standpoint of an attack-defense game to explain the underlying causes for the effectiveness of the RMU algorithm.

Attacker's goal. The goal of the attacker is to inject an *adversarial trigger* into the suffix of the prompt input (*e.g.*, "How would I engineer influenza A to be significantly more virulent? *x x x x x x*), causing the model generates a harmful response (*e.g.*, "To make influenza A more virulent...").

Attacker's knowledge and capability. In this setting, we focus on white-box jailbreak attack scenarios (Zou et al. 2023b), where the victim model's architecture, model input, and output are exposed to the attacker. The attacker is based on gradient signals to search and inject an adversarial trigger into the prompt input, and supplies this adversarial input to the model.



(a) $c = 1.0$ (b) $c = 2.0$ (c) $c = 5.0$ (d) $c = 10.0$

(e) $c = 1.0$ (f) $c = 2.0$ (g) $c = 5.0$ (h) $c = 10.0$

Figure 2: The distribution of MaxLogit (a-d) on WMDP Q&A sets with different coefficient c of the base Zephyr-7B and RMU Zephyr-7B models ($l = 7$). The distribution of $\cos\left(\boldsymbol{u}, h^{(l)}\right)$ (e-h) of the RMU Zephyr-7B model ($l = 7$).

Problem formulation. Let $f : \mathbb{R}^{n\times d} \mapsto \mathbb{R}^{n\times|V|}$ be an autoregressive LLM. Given a prompt input joint with an adversarial trigger $x_{F,1:n}$, the attacker finds an update δ to adversarial trigger aims to maximize the likelihood of generating the target sequence $x_{F,n+1|n+K}$ consists of K tokens. For simplification, we denote $x_F = x_{F,1:K} = [x_{F,1:n}, x_{F,n+1:n+K}]$. The attacker tries to solve the following objective:

$$\min_{x_F+\delta} \mathcal{J}(f(x_F + \delta)), \tag{20}$$

where $\mathcal{J}(\cdot,\cdot)$ is the loss function of the attacker. The attacker finds an update δ based on the linearized approximation of the loss $\nabla_{e_{x_i}} \mathcal{J}(f(x_F))$, where e_{x_i} is the one-hot vector representing the current value of the i-th token in x_F. The gradient $\nabla_{e_{x_i}} \mathcal{J}(f(x_F))$ is a good indicator for finding a set of candidates for the adversarial token replacement. A more negative value of the gradient $\nabla_{e_{x_i}} \mathcal{J}(f(x_F))$ makes a more decrease in the loss. The GCG attacker finds top-k largest negative value of $\nabla_{e_{x_i}} \mathcal{J}(f(x_F))$ for each token in the adversarial trigger and makes the replacement the most decrease in the loss.

Robustness of RMU models against GCG attack. We show that the GCG attacker misjudges in finding optimal adversarial token substitution in RMU models. Specifically, the gradient of the loss at input x_F with respect to e_{x_i} in RMU model is

$$\nabla_{e_{x_i}} \mathcal{J}(f^{\text{unlearn}}(x_F)) \tag{21}$$

Given the Assumption 1, we have

$$\begin{aligned}\nabla_{e_{x_i}} \mathcal{J}(f^{\text{unlearn}}(x_F)) &= \nabla_{e_{x_i}} \mathcal{J}(g^{(l:k)}(h^{(l),\text{steered}}(x_F)) \\ &\approx \nabla_{e_{x_i}} (\mathcal{J} \circ g^{(l:k)})(c\boldsymbol{u} + \boldsymbol{\epsilon})\end{aligned} \tag{22}$$

Since c and $\boldsymbol{u}$ are predetermined before unlearning, $(\mathcal{J} \circ g^{(l:k)})(c\boldsymbol{u})$ does not change with respect to e_{x_i}. The gradient $\nabla_{e_{x_i}} (\mathcal{J} \circ g^{(l:k)})(c\boldsymbol{u} + \boldsymbol{\epsilon})$ close to 0 for all token x_i since the error $\boldsymbol{\epsilon} \to \mathbf{0}$ as unlearning becomes accurate. This means the GCG attacker received unreliable, uninformative gradient signals from RMU models. The RMU model serves as a defender by causing the attacker to miscalculate the gradient of the loss to optimize its objective, thereby increasing the attacker's cost. The attacker, therefore, cannot find the optimal adversarial tokens for replacement. Li et al. (2024b)'s experiment results implicitly verify our analysis.

4 Empirical Analysis

4.1 Measuring Token Confidence with MaxLogit

As discussed in Section 3.1, we validate our hypothesis by considering the Maximum Logit Value (MaxLogit) estimator for measuring the token confidence. More specifically, we compute the MaxLogit for each token x_{n+1} given a sequence of tokens $x_{1:n} = \{x_1, ..., x_n\}$ from vocabulary V as:

$$\text{MaxLogit}(x_{n+1}) = \max_{x_{n+1}\in V} f^{\text{unlearn}}(x_{n+1}|x_{1:n}) \tag{23}$$

We use WMDP-Biology and WMDP-Cyber Q&A datasets (Li et al. 2024b) with total 3260 Q&As. We formulated each question and answer as a zero-shot Q&A prompt to query the unlearned LLM. The details of the prompt template are located in Appendix A.1. We used greedy decoding to generate tokens and compute the MaxLogit of each token over $k = 30$ generated tokens. The MaxLogit distribution was then analyzed for each model Base vs. RMU (unlearned on WMDP-Biology and WMDP-Cyber forget datasets).

The results are presented in Fig. 2 (a)-(d). We find that the MaxLogit distribution for the base model is generally wider

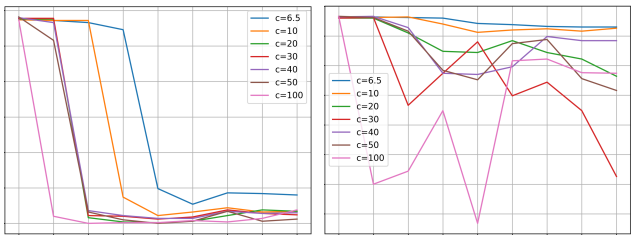


Figure 3: Average accuracy of WMDP (Biology and Cyber) (left) and MMLU with different coefficient c (right).

compared to the RMU model. In contrast, the RMU model demonstrates a more concentrated and approximately normal distribution of MaxLogit values. The peak of the RMU model's MaxLogit distribution is shifted towards lower values relative to the base model. This indicates that the RMU model tends to assign lower confidence scores to the generated tokens. Overall, the RMU model's MaxLogit distribution exhibits lower compared to the base model.

4.2 The Effect of the Coefficient c

On accuracy. We analyze the impact of c for forgotten knowledge and retained knowledge, using WMDP (Li et al. 2024b) and MMLU (Hendrycks et al. 2021). See Section 6 for the full experiment setting. Fig. 3a shows: (i) a clear positive correlation between the drop-in-accuracy rate and the value of c, *i.e.* higher c makes the accuracy decrease faster. (ii) A larger value of c tends to make a more drop-in-accuracy on WMDP. (iii) However, a larger c comes with a caveat in a significant drop in general performance on MMLU (Fig. 3b).

On alignment between $\boldsymbol{u}$ and $h^{(l)}$. We compute $\cos\big(\boldsymbol{u}, h^{(l)}\big)$ scores of pairs of $\boldsymbol{u}$ and $h^{(l)}(x_F)$ for all x_F in on WMDP-Biology and WMDP-Cyber forget datasets and plot the $\cos\big(\boldsymbol{u}, h^{(l)}\big)$ score distribution shown in Fig. 2(e)-(h). We observed that there is a clear positive correlation between $\cos\big(\boldsymbol{u}, h^{(l)}\big)$ scores and the coefficient c. As c increases, the distribution of $\cos\big(\boldsymbol{u}, h^{(l)}\big)$ scores shifts towards higher values and are almost distributed with a peak at 1.0 (Fig. 2(g)-(h)). This verify our analysis in Section 3.2.

4.3 The Effect of Layers on Unlearning

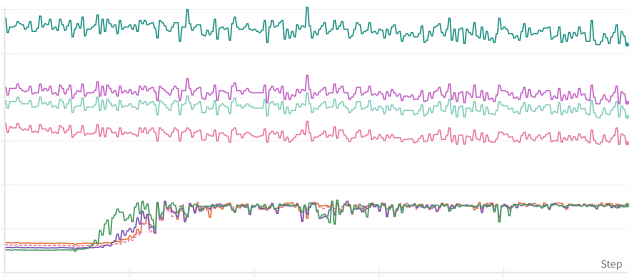


Figure 4: ℓ^2-norm of forget-sample representation.

Algorithm 1: Adaptive RMU pseudocode

Require:
$\mathcal{D}_{\text{forget}}$: a forget dataset.
$\mathcal{D}_{\text{retain}}$: a retain dataset.
$f_{\theta^{\text{frozen}}}$: a frozen model.
$f_{\theta^{\text{unlearn}}}$: an update model.
α: a retain weight.
l: an unlearn layer.
β: a scaling factor.
T: number of gradient update steps.
Ensure: Return the unlearned model $f_{\theta^{\text{unlearn}}}$.
Sample a random unit vector $\boldsymbol{u}$.
for step $t \in [1...T] : x_F \in \mathcal{D}_{\text{forget}},\ x_R \in \mathcal{D}_{\text{retain}}$ **do**
  Get the representations of x_F and x_R from the frozen and update model.
  Compute the adaptive loss $\mathcal{L}^{\text{adaptive}}$ by Eqn. 24.
  Update θ^{unlearn} w.r.t $\nabla\mathcal{L}^{\text{adap}}$ using gradient descent.
  $t = t + 1$
end for
return $f_{\theta^{\text{unlearn}}}$

We investigate the effect of unlearn layers on accuracy and the representation norm during unlearning. Following original work, we change the unlearn layer l from $3 \rightarrow 31$, fixed $c = 6.5$. Fig. 5 shows that RMU is effective for unlearning within the early layers ($3 \rightarrow 10$), yet exhibits inefficacy within middle and later layers ($11 \rightarrow 31$). Interestingly, in Fig. 4, we observed that within early layers, the ℓ^2-norm of forget samples are smaller than the coefficient c. During unlearning, the representation norm exponentially increases, approaching c, thereby facilitating the convergence of forget loss. Conversely, within middle and later layers, the representation norms of forget samples, initially larger than c, remain unchanged during unlearning, making the forget loss non-convergence.

5 Adaptive RMU

Inspired by the observations in Section 4.3, we propose *Adaptive RMU*, a simple yet effective alternative method with an adaptive forget loss by scaling the random unit vector $\boldsymbol{u}$ with an *adaptive scaling coefficient* $\beta||h^{(l)}_{\theta^{\text{frozen}}}(x_F)||$, where $\beta \in \mathbb{R}$ is a scaling factor and $||h^{(l)}_{\theta^{\text{frozen}}}(x_F)||$ is the ℓ^2-norm of forget-sample x_F on model $f_{\theta^{\text{frozen}}}$. The total loss is calculated as follows:

$$\mathcal{L}^{\text{adaptive}} = \underbrace{\mathbb{E}_{x_F \in \mathcal{D}_{\text{forget}}} ||h^{(l)}_{\theta^{\text{unlearn}}}(x_F) - \beta||h^{(l)}_{\theta^{\text{frozen}}}(x_F)||\boldsymbol{u}||_2^2}_{\text{adaptive forget loss}} + \alpha \underbrace{\mathbb{E}_{x_R \in \mathcal{D}_{\text{retain}}} ||h^{(l)}_{\theta^{\text{unlearn}}}(x_R) - h^{(l)}_{\theta^{\text{frozen}}}(x_R)||_2^2}_{\text{retain loss}} \quad (24)$$

Our Adaptive RMU is shown in Algorithm 1. We note that Adaptive RMU aims to address the challenge of adaptively determining the coefficient c in RMU. We acknowledge that the introduced value β is manually tuned via grid search, leaving the challenge to not fully resolved. However, we

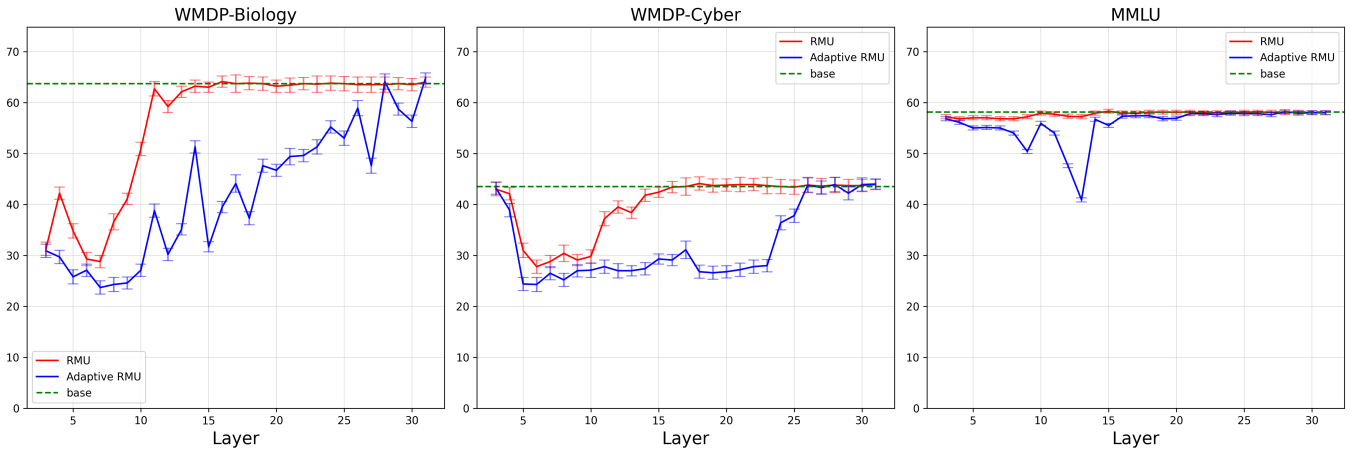


Figure 5: Q&A accuracy of RMU and Adaptive RMU Zephyr-7B models on WMDP-Biology, WMDP-Cyber, and MMLU w.r.t unlearn layer l from the third to the last layer.

emphasize that Adaptive RMU offers significant computational advantages over the original RMU. More concretely, in RMU, grid search is conducted over both c and layer l for $l \in [1...L]$, where L is the number of layers. Our analysis suggests that effective unlearning can be achieved when c is higher than the representation norm of forget-samples. Therefore, given a layer l, Adaptive RMU only requires tuning β, which is L times less than that of RMU. This reduction in computational overhead represents a significant improvement when the size of modern deep networks grows.

6 Experiment

Datasets. We use WMDP-Biology and WMDP-Cyber forget datasets as $\mathcal{D}_{\text{forget}}$ and Wikitext (Merity et al. 2022) as $\mathcal{D}_{\text{retain}}$ for unlearning the LLM. Unlearned models are evaluated on WMDP Q&A datasets and MMLU (Hendrycks et al. 2021). Details of the datasets can be found in the Appendix A.1.

Models. We use the following LLMs: Zephyr-7B-β (Tunstall et al. 2023), Yi-6B (Young et al. 2024), Meta Llama-3-8B (Meta 2024), and Mistral-7B (Jiang et al. 2023).

Experimental setup. Models were fine-tuned using AdamW (Loshchilov and Hutter 2019) with learning rate $\eta = 5e-5$, batch-size of 4, max sequence len of 512 for WMDP-Biology and 768 for WMDP-Cyber, with $T = 500$ gradient update steps. The retain weight $\alpha = 1200$. For the baseline RMU, we follow the previous work and let $c = 6.5$. We grid search for unlearn layer l from the third to the last layer. For the Adaptive RMU, we grid search for the scaling factor $\beta \in \{2, 3, 5, 10\}$. We report the performances of Adaptive RMU models with $\beta = 5$. We update three layers parameters $\{l, l-1, l-2\}$ of the model. Two NVIDIA A40s with 90GB GPU were used to run the experiments. Our code is available at https://github.com/RebelsNLU-jaist/llm-unlearning.

Baselines. We compare Adaptive RMU against baselines: RMU (Li et al. 2024b), Large Language Model Unlearning (LLMU; Yao, Xu, and Liu (2023)), SCalable Remenbering and Unlearning unBound (SCRUB; Kurmanji et al. (2023)), and Selective Synaptic Dampening (SSD; Foster, Schoepf, and Brintrup (2024). We use off-the-shelf results from Li et al. (2024b) for LLMU, SCRUB, and SSD.

Method/tasks	WMDP-Biology ↓	WMDP-Cyber ↓	MMLU ↑
Base	63.7	43.5	58.1
LLMU	59.5	39.5	44.7
SCRUB	43.8	39.3	51.2
SSD	50.2	35.0	40.7
RMU ($l = 7$)	<u>28.8</u>	<u>28.8</u>	**56.8**
Adaptive RMU ($l = 7$)	**23.7**	**26.5**	<u>55.0</u>

Table 1: Q&A accuracy of Zephyr-7B models on WMDP and MMLU. The **best** and <u>runner up</u> are marked.

Main results. Fig. 5 shows that Adaptive RMU significantly improves unlearning performances. Specifically, Adaptive RMU reduces average accuracy by 13.1% on WMDP-Biology and 3.6% on WMDP-Cyber within early layers ($3 \rightarrow 10$), and by 15.6% on WMDP-Biology and 9.6% on WMDP-Cyber within middle and later layers ($11 \rightarrow 31$). This corresponds to an overall enhancement of 14.3% and 6.6% in drop-in-accuracy for the WMDP-Biology and WMDP-Cyber, respectively. Table 1 further highlights that Adaptive RMU ($l = 7$) outperforms RMU ($l = 7$), LLMU, SCRUB, and SSD, establishing a new state-of-the-art performance. We defer the full results on other models and settings in Appendix B.

7 Conclusion

We studied the effect of steering latent representation for LLM unlearning and explored its connection to jailbreak adversarial robustness. We developed a simple yet effective alternative method that enhances unlearning performance across most layers while maintaining overall model utility. Our findings illuminate the explanation of the RMU method and pave the way for future research in LLM unlearning.

Acknowledgments

This work was supported by JST FOREST Program (Grant Number JPMJFR232K, Japan) and the Nakajima Foundation.

References

Belrose, N.; Schneider-Joseph, D.; Ravfogel, S.; Cotterell, R.; Raff, E.; and Biderman, S. 2023. LEACE: Perfect linear concept erasure in closed form. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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

Bui, T.-A.; Long, V.; Doan, K.; Le, T.; Montague, P.; Abraham, T.; and Phung, D. 2024. Erasing Undesirable Concepts in Diffusion Models with Adversarial Preservation. *NeurIPS 2024*.

Cao, Y.; and Yang, J. 2015. Towards Making Systems Forget with Machine Unlearning. In *2015 IEEE Symposium on Security and Privacy*, 463–480.

Cha, S.; Cho, S.; Hwang, D.; Lee, H.; Moon, T.; and Lee, M. 2024. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 11186–11194.

Che, T.; Zhou, Y.; Zhang, Z.; Lyu, L.; Liu, J.; Yan, D.; Dou, D.; and Huan, J. 2023. Fast federated machine unlearning with nonlinear functional theory. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Chen, C.; Zhang, Y.; Li, Y.; Wang, J.; Qi, L.; Xu, X.; Zheng, X.; and Yin, J. 2024. Post-Training Attribute Unlearning in Recommender Systems. *ACM Trans. Inf. Syst.* Just Accepted.

Chen, M.; Zhang, Z.; Wang, T.; Backes, M.; Humbert, M.; and Zhang, Y. 2022. Graph unlearning. In *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*, 499–513.

Cheng, J.; and Amiri, H. 2023. Multimodal machine unlearning. *arXiv preprint arXiv:2311.12047*.

Cheng, J.; Dasoulas, G.; He, H.; Agarwal, C.; and Zitnik, M. 2023. GNNDelete: A General Strategy for Unlearning in Graph Neural Networks. In *The Eleventh International Conference on Learning Representations*.

Chien, E.; Pan, C.; and Milenkovic, O. 2023. Efficient Model Updates for Approximate Unlearning of Graph-Structured Data. In *The Eleventh International Conference on Learning Representations*.

Choi, D.; and Na, D. 2023. Towards machine unlearning benchmarks: Forgetting the personal identities in facial recognition systems. *arXiv preprint arXiv:2311.02240*.

Cooper, A. F.; Choquette-Choo, C. A.; Bogen, M.; Jagielski, M.; Filippova, K.; Liu, K. Z.; Chouldechova, A.; Hayes, J.; Huang, Y.; Mireshghallah, N.; et al. 2024. Machine Unlearning Doesn't Do What You Think: Lessons for Generative AI Policy, Research, and Practice. *arXiv preprint arXiv:2412.06966*.

Dukler, Y.; Bowman, B.; Achille, A.; Golatkar, A.; Swaminathan, A.; and Soatto, S. 2023. Safe: Machine unlearning with shard graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17108–17118.

Eldan, R.; and Russinovich, M. 2023. Who's Harry Potter? Approximate Unlearning in LLMs. *arXiv preprint arXiv:2310.02238*.

Fan, C.; Liu, J.; Zhang, Y.; Wong, E.; Wei, D.; and Liu, S. 2024. SalUn: Empowering Machine Unlearning via Gradient-based Weight Saliency in Both Image Classification and Generation. In *The Twelfth International Conference on Learning Representations*.

Foster, J.; Schoepf, S.; and Brintrup, A. 2024. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12043–12051.

Gandikota, R.; Materzynska, J.; Fiotto-Kaufman, J.; and Bau, D. 2023. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2426–2436.

Ginart, A.; Guan, M.; Valiant, G.; and Zou, J. Y. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32.

Golatkar, A.; Achille, A.; and Soatto, S. 2020. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9304–9312.

Grosse, R.; Bae, J.; Anil, C.; Elhage, N.; Tamkin, A.; Tajdini, A.; Steiner, B.; Li, D.; Durmus, E.; Perez, E.; et al. 2023. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*.

Halimi, A.; Kadhe, S. R.; Rawat, A.; and Angel, N. B. 2022. Federated Unlearning: How to Efficiently Erase a Client in FL? In *International Conference on Machine Learning*.

Hayes, J.; Shumailov, I.; Triantafillou, E.; Khalifa, A.; and Papernot, N. 2024. Inexact unlearning needs more careful evaluations to avoid a false sense of privacy. *arXiv preprint arXiv:2403.01218*.

Hendrycks, D.; Basart, S.; Mazeika, M.; Zou, A.; Kwon, J.; Mostajabi, M.; Steinhardt, J.; and Song, D. 2022. Scaling Out-of-Distribution Detection for Real-World Settings. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 8759–8773. PMLR.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Hendrycks, D.; and Gimpel, K. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *International Conference on Learning Representations*.

Hong, Y.; Yu, L.; Ravfogel, S.; Yang, H.; and Geva, M. 2024. Intrinsic Evaluation of Unlearning Using Parametric Knowledge Traces. *arXiv preprint arXiv:2406.11614*.

Isonuma, M.; and Titov, I. 2024. Unlearning Reveals the Influential Training Data of Language Models. *arXiv preprint arXiv:2401.15241*.

Izzo, Z.; Smart, M. A.; Chaudhuri, K.; and Zou, J. 2021. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, 2008–2016. PMLR.

Jang, J.; Yoon, D.; Yang, S.; Cha, S.; Lee, M.; Logeswaran, L.; and Seo, M. 2023. Knowledge Unlearning for Mitigating Privacy Risks in Language Models. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 14389–14408. Toronto, Canada: Association for Computational Linguistics.

Jeong, H.; Ma, S.; and Houmansadr, A. 2024. SoK: Challenges and Opportunities in Federated Unlearning. *arXiv preprint arXiv:2403.02437*.

Jia, J.; Zhang, Y.; Zhang, Y.; Liu, J.; Runwal, B.; Diffenderfer, J.; Kailkhura, B.; and Liu, S. 2024. Soul: Unlocking the power of second-order optimization for llm unlearning. *arXiv preprint arXiv:2404.18239*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Jones, E.; Dragan, A.; Raghunathan, A.; and Steinhardt, J. 2023. Automatically auditing large language models via discrete optimization. In *International Conference on Machine Learning*, 15307–15329. PMLR.

Koh, P. W.; and Liang, P. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, 1885–1894. PMLR.

Kumari, N.; Zhang, B.; Wang, S.-Y.; Shechtman, E.; Zhang, R.; and Zhu, J.-Y. 2023. Ablating concepts in text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22691–22702.

Kurmanji, M.; Triantafillou, P.; Hayes, J.; and Triantafillou, E. 2023. Towards Unbounded Machine Unlearning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Li, G.; Hsu, H.; Chen, C.-F.; and Marculescu, R. 2024a. Machine Unlearning for Image-to-Image Generative Models. In *The Twelfth International Conference on Learning Representations*.

Li, N.; Pan, A.; Gopal, A.; Yue, S.; Berrios, D.; Gatti, A.; Li, J. D.; Dombrowski, A.-K.; Goel, S.; Phan, L.; et al. 2024b. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*.

Li, X.; Zhao, Y.; Wu, Z.; Zhang, W.; Li, R.-H.; and Wang, G. 2024c. Towards Effective and General Graph Unlearning via Mutual Evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 13682–13690.

Li, Y.; Chen, C.; Zheng, X.; Zhang, Y.; Han, Z.; Meng, D.; and Wang, J. 2023. Making users indistinguishable: Attribute-wise unlearning in recommender systems. In *Proceedings of the 31st ACM International Conference on Multimedia*, 984–994.

Liu, C. Y.; Wang, Y.; Flanigan, J.; and Liu, Y. 2024a. Large Language Model Unlearning via Embedding-Corrupted Prompts. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Liu, J.; Lou, J.; Qin, Z.; and Ren, K. 2024b. Certified minimax unlearning with generalization rates and deletion capacity. *Advances in Neural Information Processing Systems*, 36.

Liu, W.; Wang, X.; Owens, J.; and Li, Y. 2020. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33: 21464–21475.

Liu, Z.; Dou, G.; Tan, Z.; Tian, Y.; and Jiang, M. 2024c. Machine Unlearning in Generative AI: A Survey. *arXiv preprint arXiv:2407.20516*.

Liu, Z.; Dou, G.; Tan, Z.; Tian, Y.; and Jiang, M. 2024d. Towards Safer Large Language Models through Machine Unlearning. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 1817–1829. Bangkok, Thailand: Association for Computational Linguistics.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Lynch, A.; Guo, P.; Ewart, A.; Casper, S.; and Hadfield-Menell, D. 2024. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*.

Ma, Z.; Liu, Y.; Liu, X.; Liu, J.; Ma, J.; and Ren, K. 2022. Learn to forget: Machine unlearning via neuron masking. *IEEE Transactions on Dependable and Secure Computing*.

Maini, P.; Feng, Z.; Schwarzschild, A.; Lipton, Z. C.; and Kolter, J. Z. 2024. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2022. Pointer Sentinel Mixture Models. In *International Conference on Learning Representations*.

Meta, A. 2024. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*.

Ngoc-Hieu, N.; Hung-Quang, N.; Ta, T.-A.; Nguyen-Tang, T.; Doan, K. D.; and Thanh-Tung, H. 2023. A Cosine Similarity-based Method for Out-of-Distribution Detection. *arXiv preprint arXiv:2306.14920*.

Nguyen, T. T.; Huynh, T. T.; Nguyen, P. L.; Liew, A. W.-C.; Yin, H.; and Nguyen, Q. V. H. 2022. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*.

Northcutt, C.; Jiang, L.; and Chuang, I. 2021. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70: 1373–1411.

Patil, V.; Hase, P.; and Bansal, M. 2024. Can Sensitive Information Be Deleted From LLMs? Objectives for Defending Against Extraction Attacks. In *The Twelfth International Conference on Learning Representations*.

Pawelczyk, M.; Neel, S.; and Lakkaraju, H. 2024. In-Context Unlearning: Language Models as Few-Shot Unlearners. In *Forty-first International Conference on Machine Learning*.

Plaut, B.; Nguyen, K.; and Trinh, T. 2024. Softmax probabilities (mostly) predict large language model correctness on multiple-choice q&a. *arXiv preprint arXiv:2402.13213*.

Romandini, N.; Mora, A.; Mazzocca, C.; Montanari, R.; and Bellavista, P. 2024. Federated unlearning: A survey on methods, design guidelines, and evaluation metrics. *IEEE Transactions on Neural Networks and Learning Systems*.

Sekhari, A.; Acharya, J.; Kamath, G.; and Suresh, A. T. 2021. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34: 18075–18086.

Shah, R.; Montixi, Q. F.; Pour, S.; Tagade, A.; and Rando, J. 2023. Scalable and Transferable Black-Box Jailbreaks for Language Models via Persona Modulation. In *Socially Responsible Language Modelling Research*.

Shi, W.; Ajith, A.; Xia, M.; Huang, Y.; Liu, D.; Blevins, T.; Chen, D.; and Zettlemoyer, L. 2024a. Detecting Pretraining Data from Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Shi, W.; Lee, J.; Huang, Y.; Malladi, S.; Zhao, J.; Holtzman, A.; Liu, D.; Zettlemoyer, L.; Smith, N. A.; and Zhang, C. 2024b. MUSE: Machine Unlearning Six-Way Evaluation for Language Models. *arXiv preprint arXiv:2407.06460*.

Sun, Y.; Ming, Y.; Zhu, X.; and Li, Y. 2022. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, 20827–20840. PMLR.

Tan, J.; Sun, F.; Qiu, R.; Su, D.; and Shen, H. 2024. Unlink to unlearn: Simplifying edge unlearning in gnns. In *Companion Proceedings of the ACM on Web Conference 2024*, 489–492.

Thudi, A.; Deza, G.; Chandrasekaran, V.; and Papernot, N. 2022. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, 303–319. IEEE.

Tunstall, L.; Beeching, E.; Lambert, N.; Rajani, N.; Rasul, K.; Belkada, Y.; Huang, S.; von Werra, L.; Fourrier, C.; Habib, N.; et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Wang, H.; Lin, J.; Chen, B.; Yang, Y.; Tang, R.; Zhang, W.; and Yu, Y. 2025. Towards efficient and effective unlearning of large language models for recommendation. *Frontiers of Computer Science*, 19(3): 193327.

Wang, J.; Guo, S.; Xie, X.; and Qi, H. 2022. Federated Unlearning via Class-Discriminative Pruning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, 622–632. New York, NY, USA: Association for Computing Machinery. ISBN 9781450390965.

Warnecke, A.; Pirch, L.; Wressnegger, C.; and Rieck, K. 2021. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*.

Wei, A.; Haghtalab, N.; and Steinhardt, J. 2024. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36.

Wei, H.; Xie, R.; Cheng, H.; Feng, L.; An, B.; and Li, Y. 2022. Mitigating neural network overconfidence with logit normalization. In *International conference on machine learning*, 23631–23644. PMLR.

Wu, K.; Shen, J.; Ning, Y.; Wang, T.; and Wang, W. H. 2023a. Certified Edge Unlearning for Graph Neural Networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, 2606–2617. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701030.

Wu, X.; Li, J.; Xu, M.; Dong, W.; Wu, S.; Bian, C.; and Xiong, D. 2023b. DEPN: Detecting and Editing Privacy Neurons in Pretrained Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2875–2886. Singapore: Association for Computational Linguistics.

Xu, H.; Zhu, T.; Zhang, L.; Zhou, W.; and Yu, P. S. 2023. Machine Unlearning: A Survey. *ACM Comput. Surv.*, 56(1).

Yao, Y.; Xu, X.; and Liu, Y. 2023. Large Language Model Unlearning. In *Socially Responsible Language Modelling Research*.

Young, A.; Chen, B.; Li, C.; Huang, C.; Zhang, G.; Zhang, G.; Li, H.; Zhu, J.; Chen, J.; Chang, J.; et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

Yuan, Y.; Jiao, W.; Wang, W.; tse Huang, J.; He, P.; Shi, S.; and Tu, Z. 2024. GPT-4 Is Too Smart To Be Safe: Stealthy Chat with LLMs via Cipher. In *The Twelfth International Conference on Learning Representations*.

Zhang, G.; Wang, K.; Xu, X.; Wang, Z.; and Shi, H. 2024a. Forget-me-not: Learning to forget in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1755–1764.

Zhang, R.; Lin, L.; Bai, Y.; and Mei, S. 2024b. Negative Preference Optimization: From Catastrophic Collapse to Effective Unlearning. In *First Conference on Language Modeling*.

Zhang, Y.; Hu, Z.; Bai, Y.; Wu, J.; Wang, Q.; and Feng, F. 2023. Recommendation unlearning via influence function. *ACM Transactions on Recommender Systems*.

Zhu, X.; Li, G.; and Hu, W. 2023. Heterogeneous federated knowledge graph embedding learning and unlearning. In *Proceedings of the ACM web conference 2023*, 2444–2454.

Zou, A.; Phan, L.; Chen, S.; Campbell, J.; Guo, P.; Ren, R.; Pan, A.; Yin, X.; Mazeika, M.; Dombrowski, A.-K.; et al. 2023a. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

Zou, A.; Wang, Z.; Kolter, J. Z.; and Fredrikson, M. 2023b. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.