Master's Thesis

Understanding the Effects of Representation Misdirection for Large Language Model Unlearning

Huu-Tien Dang

Supervisor: Associate Professor Naoya Inoue

Examiners: Professor Okada Shogo
Professor Le-Minh Nguyen
Professor Kiyoaki Shirai

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Information Science)

August, 2025

Abstract

Modern large language models (LLMs) are pre-trained on massive text corpora from the web, including copyrighted material, toxic and sexual content, sensitive and private information, gender and political bias, and dangerous documents such as cybersecurity attacks and bioweapon development. As a result, LLMs can exhibit harmful and unwanted behaviors. The right to be forgotten emerged for LLMs as a tool to ensure their and our safety. Machine unlearning is an approach that aims to *remove* or *suppress* the target forget knowledge and capabilities from a pre-trained model while maintaining the model's other knowledge and capabilities. Representation Misdirection for Unlearning (RMU)—an approach that performs unlearning by manipulating the latent representations of the forget-samples in the pre-trained models—establishes LLM unlearning methods with state-of-the-art performance. Yet, the underlying causes and explanations remain underexplored.

In this thesis, we show the following: (1) a theoretical analysis that demonstrates steering the forget-representations to a target random representation of RMU's objective reduces forget-token confidence, causing LLMs to generate wrong or nonsense responses. (2) How the coefficient influences the alignment of forget-representations with the random direction and hints at the optimal coefficient values for effective unlearning across different network layers. (3) Analyzing RMU's robustness against white-box knowledge recovery attacks through the lens of an attack–defense game: RMU acts as a defender, impeding the adversary's ability to determine optimal updates for generating adversarial samples, thus improving the adversarial robustness of unlearned models. (4) RMU's forget-loss, which minimizes the mean squared error 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 deep layers in LLMs. To overcome this limitation, we introduce *Adaptive RMU*, a simple yet effective alternative to RMU, that adaptively adjusts the coefficient based on the norm of forget-representations. Extensive experiments demonstrate that Adaptive RMU achieves higher drop-in accuracy for forgetting knowledge, maintains high performance on general knowledge, and enables effective unlearning for most layers without incurring additional computational overhead. Our implementation is available at https://github.com/huutiendang/llm-unlearning.

Keywords—Machine Unlearning, Adversarial Robustness, Large Language Model, Representation Misdirection for Unlearning.

Published Works and AI Usage Declaration

I hereby declare that, to my best knowledge and belief, this thesis contains no material previously published or written by another person, except where due references are made in the text of the thesis.

AI tools were used for editing and formatting purposes (*e.g.*, grammar checking). **No AI tools were used for implementing the method and experimental setup**. The ideas and development of this thesis were my responsibility.

This thesis's content is based on: one paper published in a peer-reviewed conference and one submitted to a peer-reviewed conference. The theme of the thesis is analyzing the effects of representation misdirection for unlearning and developing efficient machine unlearning algorithms.

Publications

1. 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.

2. Huu-Tien Dang, Hoang Thanh-Tung, Anh Bui, Le-Minh Nguyen, and Naoya Inoue. Improving LLM Unlearning Robustness via Random Perturbations. arXiv preprint arXiv:2501.19202, 2025.

Student: Huu-Tien Dang
Date: 08/2025

Acknowledgment

To my dear family and friends: This dissertation is dedicated to them.

Contents

List of Figures

List of Tables

Abbreviations

Adaptive RMU	Adaptive Representation Misdirection for Unlearning
AI	Artificial Intelligence
API	Application Programming Interface
CCPA	California Consumer Privacy Act
DPO	Direct Preference Optimization
FLAT	Forget data only loss adjustment
GCG	Greedy Coordinate Gradient
GDPR	General Data Protection Regulation
GPT	Generative Pre-trained Transformer
LLM	Large Language Model
LLMU	Large Language Model Unlearning
MaxLogit	Maximum Logit
MMLU	Massive Multitask Language Understanding
MU	Machine Unlearning
NPO	Negative Preference Optimization
RepE	Representation Engineering
RLHF	Reinforcement Learning from Human Feedback
RMU	Representation Misdirection for Unlearning
SCRUB	Scalable Remembering and Unlearning unBound
SimNPO	Simple Negative Preference Optimization
SSD	Selective Synaptic Dampening
STEM	Science, Technology, Engineering, and Mathematics
WMDP	Weapon of Mass Destruction Proxy

Chapter 1

Introduction

1.1 Motivation: AI is Dual-use

> *"The rapid progress in AI comes with many short-term risks. It has already created divisive echo chambers by offering people content that makes them indignant. It is already being used by authoritarian governments for massive surveillance and by cyber-criminals for phishing attacks. In the near future, AI may be used to create terrible new viruses and horrendous lethal weapons that decide by themselves who to kill or maim."*
>
> — Geoffrey Hinton, Nobel Prize in Physics 2024 —

Technologies such as drones, nuclear energy, gene editing, etc., share a *dual-use* nature: they can be leveraged for social good or, if catastrophically misused, can cause significant harm. AI is similar. From coding (Anthropic, 2024), discovering new materials (Merchant et al., 2023), predicting new protein structures (Abramson et al., 2024), solving hard math problems in Euclidean geometry (Trinh et al., 2024), to generating scientific papers that have successfully passed peer review at the scientific workshop (Yamada et al., 2025), and even the A* scientific main conference[1]—AI is rapidly transforming the landscape of basic life, research, and innovation.

Alongside these amazing capabilities, AI—especially in the era of LLMs—also poses serious risks. LLMs can (adversarially) generate copyrighted materials (Eldan and Russinovich, 2024; Karamolegkou et al., 2023; He et al., 2024; Shi et al., 2025), perpetuate unfair treatment, gender, and language bias (Belrose et al.,

[1] https://www.intology.ai/blog/zochi-acl

2023), produce hallucinations (Huang et al., 2025; Zhang et al., 2023b), enable data poisoning, cybersecurity attacks (Fang et al., 2024). They can also be used for developing destructive chemical and biological weapons (Sandbrink, 2023; Li et al., 2024b).

As the size and capabilities of LLMs continue to grow at unprecedented speed, it becomes increasingly urgent to develop safeguards and norms to ensure that LLMs are safe, that is, trustworthy, reliable, and secure.

Mitigating the risks from LLMs. To mitigate the LLM risks, many strategies have been proposed, such as Reinforcement Learning from Human Feedback (RLHF; Christiano et al. (2017); Ziegler et al. (2019); Stiennon et al. (2020); Ouyang et al. (2022)), Direct Preference Optimization (DPO Rafailov et al. (2023)), and Representation Engineering (RepE; Zou et al. (2023a)). However, researchers showed that even though LLMs are post-trained with these alignment methods to be helpful and harmless (Bai et al., 2022), well-aligned LLMs remain susceptible to adversarial jailbreak attacks, which can bypass safeguards and elicit harmful, unwanted behaviors and outputs (Wei et al., 2023; Chao et al., 2025; Zou et al., 2023b).

Machine unlearning (MU). MU (Cao and Yang, 2015; Nguyen et al., 2022; Xu et al., 2023), a post-training approach, has emerged as a promising way to teach the models "forget" or "unlearn" undesirable behaviors, data, or harmful capabilities while maintaining general ones. We note that this is an informal definition of MU; depending on the context, definition, goals, and evaluation of MU might take different forms. We defer the discussion on related works to Section 2.

Representation Misdirection for Unlearning (RMU; Li et al. (2024b)) is the state-of-the-art unlearning approach that achieves unlearning by steering the forget-representations (*i.e.,* latent representations of forget-tokens) toward a random target representation while keeping the retain-representations (*i.e.,* latent representations of retain-tokens) remain unchanged. RMU significantly degrades models' accuracy on forget-tasks, while slightly affecting the accuracy on retain-tasks, and demonstrates stronger robustness against adversarial jailbreak attacks. We defer the details of RMU's formulation to Section 2.2.

However, the reasons behind RMU's effectiveness are not well understood, which hinders the development of better MU algorithms.

1.2 Aims and Scope

This thesis aims to analyze the effects of Representation Misdirection for Unlearning in autoregressive LLMs, which is the most widely used and state-of-the-art unlearning method to date. As we will show later, RMU is linked to confidence estimation, adversarial robustness, and noise stability of the models. We conduct an extensive empirical analysis and develop an alternative version of RMU, which improves the RMU's efficiency. Finally, we discuss the ongoing debate surrounding the underlying mechanisms of LLM unlearning and consider a critical question: Do existing LLM unlearning methods truly erase target knowledge and harmful behaviors, or do they try to suppress them from occurring (as humans often do!)?

1.3 Contributions

This thesis is an attempt to answer the following general research questions about RMU for LLM unlearning. The questions are:

1. What are the causes of RMU's effectiveness and robustness?
2. What are the limitations of RMU, and how to fix them?
3. Does RMU truly unlearn?

To this end, we have found partial answers to the above questions. We make the following contributions:

1. **Behavioral Effects:** We show that steering the forget-representations in the intermediate layers in LLM will reduce token confidence, causing the LLMs to generate wrong or nonsense answers.
2. **Adversarial Robustness**: We show that RMU impedes the adversary's ability to find optimal updates for generating adversarial samples, thus improving the RMU models' adversarial robustness.
3. **Limitations:** We analyze the effects of the coefficient on RMU forget-loss and found that RMU forget-loss, which minimizes the mean squared error between forget-representations and a random scaled vector, *fails* to converge when the norm of the forget-representation is larger than the coefficient, making RMU ineffective when applied to deep layers of the LLMs. To address this limitation, we introduce *Adaptive RMU*, a variant that adaptively adjusts the coefficient based on the norm of forget-representations. Extensive experiments show that Adaptive RMU enables effective unlearning for most layers without incurring additional computational overhead.

4. **Mechanism**: We propose a novel perspective that views *RMU as a backdoor attack process*. We formulate how the RMU learns to align forget-tokens (backdoor triggers) with the target random representation (the target label). As a result, RMU models would behave like they don't know when asked with forget queries. Further, this backdoor formulation unveils the vulnerability of RMU unlearned models; that is, when the forget-tokens appear in the retain queries, the model will misbehave. Thus, RMU itself reduces the robustness and generalization of the models.

1.4 Structure

In the next chapter, we briefly discuss the definition, formulation, goals, targets, and current state-of-the-art methods of unlearning. We then review RMU, the most widely used method for LLMs to date.

Chapter 3 presents a comprehensive analysis of RMU. We discuss the effects of RMU on the confidence of generated tokens, the effects of the coefficient on the accuracy and alignment between forget-representations and the random direction. Next, we formulate the RMU from the perspective of noise sensitivity, hence unveiling the optimal coefficient for effective unlearning across layers. In the last part of this chapter, we explain the RMU's adversarial robustness from the view of an attack-defense game.

Chapter 4 presents empirical analysis for the theoretical analysis of RMU.

Chapter 5 introduces Adaptive RMU, a simple yet effective variant of RMU.

Chapter 6 compares existing unlearning methods vs. Adaptive RMU and discusses our perspective on the underlying mechanism of RMU.

The final chapter summarizes the thesis's content, discusses limitations, open problems, and future directions.

Chapter 2

Background and Related Work

In this chapter, we introduce the basic background and review the related works.

Notation. We first define the general notations used in this thesis. We denote matrices by boldface uppercase letters (*e.g.,* $\boldsymbol{A}$, $\boldsymbol{B}$), vectors by boldface lowercase letters (*e.g.,* $\mathbf{x}$, $\mathbf{y}$), and scalars (real numbers) by lowercase letters (*e.g.,* c, d, α, β). $||\cdot||$ denotes the Euclidean norm. For operators, we denote $\circ$ the decomposition operator (*e.g.,* $f = g \circ h$ mean function f can be decomposed into 2 parts: g and h).

Let $\mathcal{D}_f = \{(\mathbf{x}^f, \mathbf{y}^f)\}_i$ be the forget-set, where $\mathbf{x}^f$ is the forget-input and $\mathbf{y}^f$ is the target forget output. $\mathcal{D}_r = \{(\mathbf{x}^r, \mathbf{y}^r)\}_j$ denotes the retain-set, where $\mathbf{x}^r$ is the retain-input and $\mathbf{y}^r$ is the target retain-output. Let $f_{\boldsymbol{\theta}} : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times |V|}$ be an autoregressive LLM parameterized by $\boldsymbol{\theta}$ that maps the input $\mathbf{x} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ consisting of n tokens to an output of probability distributions over the vocabulary V. Let $\ell(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})$ be the loss of output $\mathbf{y}$ given input $\mathbf{x}$ in model $f_{\boldsymbol{\theta}}$.

Denote $h^{(l)}_{\boldsymbol{\theta}}(\mathbf{x}^f) \in \mathbb{R}^{n \times d_l}$, $h^{(l)}_{\boldsymbol{\theta}}(\mathbf{x}^r) \in \mathbb{R}^{n \times d_l}$ the output hidden state of forget-input and retain-input, respectively, at the intermediate layer l in model $f_{\boldsymbol{\theta}}$, where d_l is the dimension of layer l. For simplicity, without other clarifications, we use $h^{(l)}(\mathbf{x}^f)$ to present $h^{(l)}_{\boldsymbol{\theta}}(\mathbf{x}^f)$.

2.1 Machine Unlearning

2.1.1 Problem Formulation

The training data of a machine unlearning problem consists of two subsets: the forget-set $\mathcal{D}_f = \{(\mathbf{x}^f, \mathbf{y}^f)\}_i$ and the retain-set $\mathcal{D}_r = \{(\mathbf{x}^r, \mathbf{y}^r)\}_j$. The goal is to minimize the model's performance on the forget-set while keeping the performance

on the retain-set. A commonly used form of unlearning involves minimizing the following two-part loss:

$$\mathcal{L}_{\mathcal{D}_f,\mathcal{D}_r,\boldsymbol{\theta}} = \mathbb{E}_{(\mathbf{x}^f,\mathbf{y}^f)\sim\mathcal{D}_f}\left[\ell\left(\mathbf{y}^f|\mathbf{x}^f;\boldsymbol{\theta}\right)\right] + \alpha\mathbb{E}_{(\mathbf{x}^r,\mathbf{y}^r)\sim\mathcal{D}_r}\left[\ell\left(\mathbf{y}^r|\mathbf{x}^r;\boldsymbol{\theta}\right)\right] \tag{2.1}$$

Note that some works utilize custom losses for specific purposes, *e.g.,* Yao et al. (2024) use the Random Mismatch loss—a regularization term to control the magnitude of parameter updates.

2.1.2 Related Work

Machine unlearning. Recent legislation on right-to-be-forgotten (Shastri et al., 2019), such as the General Data Protection Regulation (GDPR; Mantelero (2013)) and the California Consumer Privacy Act (CCPA; BUKATY (2019)) has raised attentions to a new learning paradigm called machine unlearning (MU; Cao and Yang (2015); Bourtoule et al. (2021); Chris Jay Hoofnagle and Borgesius (2019); Nguyen et al. (2022)), an approach can be broadly described as removing unwanted behaviors or data from pre-trained models.

MU for specific domains, tasks, and settings. The landscape of MU has rapidly expanded to encompass a diverse range of domains, tasks, and settings. In text, MU has been applied to text classification (Ma et al., 2022), in the vision domain, efforts include unlearning in image classification and recognition (Ginart et al., 2019; Golatkar et al., 2020; Fan et al., 2024b; Choi and Na, 2023; Cha et al., 2024), image-to-image generative models (Li et al., 2024a).

MU has also been extended to 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 et al., 2024), graph unlearning (Chen et al., 2022; Chien et al., 2023; Wu et al., 2023a; Cheng et al., 2023; Dukler et al., 2023; Zhu et al., 2023; Li et al., 2024c; Tan et al., 2024), recommender systems (Zhang et al., 2023a; Chen et al., 2024; Li et al., 2023; Wang et al., 2025a).

Beyond application domains, recent efforts have also focused on certified and minimax unlearning guarantees (Liu et al., 2024a), unlearning of specific types of targets and information (Cooper et al., 2024), and comprehensive evaluation methodologies for assessing unlearning effectiveness and robustness (Lynch et al., 2024; Hayes et al., 2024; Shi et al., 2024a; Wu et al., 2024; Shi et al., 2024b; Wei et al., 2025; Scholten et al., 2024).

MU for LLMs. Recent research on MU for LLMs has largely focused on task- or context-specific scenarios. These include copyrighted material, such as content from the Harry Potter series (Eldan and Russinovich, 2023), in-context unlearning Pawelczyk et al. (2024), and fictitious unlearning (Maini et al., 2024). Other works target the removal of specific harmful input–output (Yao et al., 2023; Liu et al., 2024b), sensitive or private information (Jang et al., 2023; Wu et al., 2023b; Patil et al., 2024), gender bias (Belrose et al., 2023), and concept-level aware (Hong et al., 2024). Most recently, Li et al. (2024b) introduced a benchmark and framework for unlearning an entire distribution of hazardous knowledge, extending the scope of MU beyond individual samples.

MU algorithms for LLMs. We broadly categorize current LLM unlearning methods into two widely used classes, including: (1) Preference Optimization (PO) based, such as Negative Preference Optimization (NPO; Zhang et al. (2024b)) (NPO is a generalized version of Gradient Ascent), Simple Negative Preference Optimization (SimNPO; Fan et al. (2024a)) a simplified and more efficient variant of NPO, and Direct Preference Optimization (DPO; Rafailov et al. (2023)) and (2) Representation Misdirection for Unlearning (RMU; Li et al. (2024b)), which aims to steer internal representations away from forget-specific information. Additionally, other notable variants include Maximum Entropy (Yuan et al., 2025), forget data only loss adjustment (FLAT; Wang et al. (2025b)).

Unlearning robustness in LLMs. Emerging studies highlight a critical challenge in LLM unlearning: **the fragility of unlearned models**. Despite successful unlearning interventions, forgotten information can resurface in various ways. These include relearning (Li et al., 2024b; Deeb and Roger, 2025; Łucki et al., 2024), sequential unlearning (Shi et al., 2025), target relearning attacks (Hu et al., 2025), steering specific directions in the latent space (Łucki et al., 2024), model quantization (Zhang et al., 2025), or even simply fine-tuning on unrelated tasks (Łucki et al., 2024). As these findings suggest, unlearning robustness against knowledge recovery is now a central concern in the design of effective MU algorithms.

2.2 Representation Misdirection for Unlearning

RMU (Algorithm 1; Li et al. (2024b)) is a fine-tuning based unlearning method inspired by representation engineering (Zou et al., 2023a). RMU optimizes 2 objectives. First, RMU pushes the forget-representations at an intermediate layer l, denoted as $h_{\boldsymbol{\theta}}^{(l)}(\mathbf{x}^f)$, toward a predefined random representation $\mathbf{y}^f = c\mathbf{u}$, where $\mathbf{u} \in \mathbb{R}^{d_l}$ is a random unit vector, each element is uniformly sampled from $[0, 1)$, and $c \in \mathbb{R}_+$ is a forget coefficient. Second, RMU regularizes the retain-representations

Algorithm 1 RMU pseudocode

Require:
A forget dataset $\mathcal{D}_r$, a retain dataset $\mathcal{D}_r$, a reference (frozen weight) model $f_{\theta^{\text{ref}}}$, an unlearn (update) model f_θ, a retain weight α, an unlearn layer l, a forget coefficient c, number of update step T.

Ensure: Return the unlearned model f_θ.

Sample a random vector $\mathbf{u}$, where each entry drawn uniformly from $[0, 1)$.
for step $t \in [1...T] : \mathbf{x}^f \sim \mathcal{D}_f,\ \mathbf{x}^r \sim \mathcal{D}_r$ **do**
    Forward hook the representations of $\mathbf{x}^f$ and $\mathbf{x}^r$ from the frozen and update model.
    Compute the loss $\mathcal{L}^{\text{RMU}}$ by Eqn. 2.2.
    Update θ w.r.t $\nabla_\theta \mathcal{L}^{\text{RMU}}$ using stochastic gradient descent.
    $t = t + 1$
end for
return f_θ

of the update model, denoted as $h^{(l)}_{\boldsymbol{\theta}}(\mathbf{x}^r)$, back to the reference model's retain-representations $h^{(l)}_{\boldsymbol{\theta}^{\text{ref}}}(\mathbf{x}^r)$. The total loss of RMU is defined as

$$\mathcal{L}^{\text{RMU}}_{\boldsymbol{\theta},\boldsymbol{\theta}^{\text{ref}},\mathcal{D}_f,\mathcal{D}_r} = \mathbb{E}_{\mathbf{x}^f \sim \mathcal{D}_f} ||h^{(l)}_{\boldsymbol{\theta}}(\mathbf{x}^f) - c\mathbf{u}||^2 + \alpha \mathbb{E}_{\mathbf{x}^r \sim \mathcal{D}_r} ||h^{(l)}_{\boldsymbol{\theta}}(\mathbf{x}^r) - h^{(l)}_{\boldsymbol{\theta}^{\text{ref}}}(\mathbf{x}^r)||^2, \quad (2.2)$$

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

RMU has been empirically demonstrated to be able to drop the accuracy of forgotten knowledge to near random without crippling model performance on general knowledge and show strong robustness against state-of-the-art white box adversarial jailbreak attack. However, many critical questions about RMU remain unanswered. In Chapter 3, we present a comprehensive analysis of RMU.

Chapter 3

Analysis of RMU

3.1 Token Confidence in RMU Models

In general, data points from shifted distributions, such as out-of-distribution, noisy labels, or poisoned data, are associated with lower confidence scores, such as maximum softmax probability (Hendrycks and Gimpel, 2017; Northcutt et al., 2021), maximum logit score (Hendrycks et al., 2022; Wei et al., 2022), Euclidean distance Sun et al. (2022), cosine similarity (Ngoc-Hieu et al., 2023), or energy score (Liu et al., 2020).

Building upon previous works, we hypothesize that the output representation of generated tokens in RMU models exhibits randomness. As seen by a deep neural network, such randomness might lower the confidence of the output representation of generated tokens, resulting in incorrect answers. To validate the hypothesis, we first make the following definition and assumption.

Assumptions and Definitions

Definition 1. *(Unlearned model). Suppose that L-layer model f can be decomposed into $f = g \circ h^{(l)}$, where g is the transformation from layer l to the last layer of network f, for any layer $l \in [1...L]$. We define the unlearned model $f^u = g \circ h^{(l),rand}$, $h^{(l),rand}$ is the randomized representation of the given input at layer l. The output representation of the next token $\mathbf{x}_n^f$ given a sequence of forget-tokens $\mathbf{x}_{<n}^f$, obtained from f^u is defined as:*

$$f^u(\mathbf{x}^f|\mathbf{x}_{<n}^f) = (g \circ h^{(l),rand})(\mathbf{x}_n^f|\mathbf{x}_{<n}^f) \tag{3.1}$$

$$= g(h^{(l),rand}(\mathbf{x}_n^f|\mathbf{x}_{<n}^f)) \tag{3.2}$$

Assumption 1. *(Randomized representation) A well-unlearned model shifts the forget-representations to a scaled random vector $c\mathbf{u}$. More concretely,*

$$h^{(l),rand}(\mathbf{x}_i^f) = c\mathbf{u} + \boldsymbol{\epsilon}, \tag{3.3}$$

where $\mathbf{x}_i^f$ is the i-th token in $\mathbf{x}_{<n}^f$, $\boldsymbol{\epsilon}$ is a small error. Without loss of generality, we assume that $\boldsymbol{\epsilon}$ is sampled from Normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. where $\boldsymbol{\Sigma} = \eta \boldsymbol{I}$ is the covariance matrix, $\eta \in \mathbb{R}_+$.

Proposition 1. *If Assumption 1 holds, by Definition 1, the output representation of forget token $\mathbf{x}_n^f$ given the previous sequence $\mathbf{x}_{<n}^f$ generated by model f^u, given as $f^u(\mathbf{x}_n^f|\mathbf{x}_{<n}^f)$ follows the Normal distribution $\mathcal{N}\left(g(\mathbf{z}), \eta \boldsymbol{J}^\top \boldsymbol{J}\right)$, where $\mathbf{z} = c\mathbf{u}$ and $\boldsymbol{J} = \nabla_{\mathbf{z}} g(\mathbf{z})$ is the Jacobian of g at $\mathbf{z}$.*

Proof. Assumption 1 implies that in a well-unlearned model, token $\mathbf{x}_n^f$ is independent of the previous tokens $x_{<n}^f$, thus we have:

$$h^{(l),\text{rand}}(\mathbf{x}_n^f|\mathbf{x}_{<n}^f) = h^{(l),\text{rand}}(\mathbf{x}_n^f) = c\mathbf{u} + \boldsymbol{\epsilon} \tag{3.4}$$

Denote $\mathbf{z} = c\mathbf{u}$. Substituting Eqn. 3.4 into Eqn. 3.2, we get:

$$f^{\text{u}}(\mathbf{x}_n^f|\mathbf{x}_{<n}^f) = g(\mathbf{z} + \boldsymbol{\epsilon}) \tag{3.5}$$

Since $\boldsymbol{\epsilon}$ is small, we approximate the function $g(\mathbf{z} + \boldsymbol{\epsilon})$ using the first-order Taylor approximation:

$$f^u(x_n^f|x_{<n}^f) \approx g(\mathbf{z}) + \nabla_{\mathbf{z}} g(\mathbf{z})^\top \boldsymbol{\epsilon} \tag{3.6}$$

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

$$f^u(x_n^f|x_{<n}^f) \sim \mathcal{N}\left(g(\mathbf{z}), \eta \nabla_{\mathbf{z}} g(\mathbf{z})^\top \nabla_{\mathbf{z}} g(\mathbf{z})\right) \tag{3.7}$$

□

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

Proposition 1 suggests that the variance of $f^u(x_n^f|x_{<n}^F)$ is controlled by (i) η: a scalar variance and (ii) the matrix $\boldsymbol{J}^\top \boldsymbol{J}$. If $f^u(x_n^f|x_{<n}^f)$ has high variance, the output is more random. Since $\boldsymbol{\epsilon}$ presents a small error, then $\boldsymbol{\epsilon}$ varies for different inputs. This variation makes it difficult to control the variance of the output by η. The main effect depends on $\boldsymbol{J}^\top \boldsymbol{J}$. The product $\boldsymbol{J}^\top \boldsymbol{J}$ varies depending on the specific characteristics of sub-networks g and input $\mathbf{z} = c\mathbf{u}$. Unfortunately,

g is a composition of transformer layers, which is nonlinear, making it difficult to have a complete analysis. The variance of $\mathbf{z}$, derived as $\text{Var}(\mathbf{z}) = c^2\text{Var}(\mathbf{u})$, is proportional to c; *i.e.* when c gets larger, the variance of $\mathbf{z}$ is higher. This could increase the variability of $g(\mathbf{z})$ and the gradient $\nabla_{\mathbf{z}} g(\mathbf{z})$. *A larger c could introduces more randomness to the output.* We conduct an empirical analysis to understand the confidence of generated tokens by RMU models in Section 6.

3.2 The Effects of the Coefficient

RMU forget-loss steers the forget-representation $h^{(l)}(\mathbf{x}^f)$ aligns with a random direction given by random unit vector $\mathbf{u}$ and scales the magnitude of $h^{(l)}(\mathbf{x}^f)$ to c. While $\mathbf{u}$ is predetermined before unlearning, the magnitude of $h^{(l)}(\mathbf{x}^f)$ varies depending on the input $\mathbf{x}^f$ and specific properties of unlearn layer l. One might ask the following research questions:

1. RQ1 (Direction): "How does the coefficient c influence the alignment between $h^{(l)}(\mathbf{x}^f)$ with $\mathbf{u}$?"

2. RQ2 (Magnitude): "What is the optimal coefficient c for effectively unlearning with different layers?"

We aim to answer RQ1 and RQ2 by analyzing the machine unlearning problem from a *compression perspective.*

We consider the output of f given input $\mathbf{x}$: $f(\mathbf{x}) = g(h^{(l)}(\mathbf{x}))$. For simplicity, in this section, without any clarifications about the layer, we write $h(\mathbf{x})$ to present $h^{(l)}(\mathbf{x})$. Suppose that we compress a random vector $\boldsymbol{\xi}$ to representation $h(\mathbf{x})$, then the output becomes $f(\mathbf{x}) = g(h(\mathbf{x}) + \boldsymbol{\xi})$. Naturally, if g is robust (less sensitive) to noise $\boldsymbol{\xi}$, then $\boldsymbol{\xi}$ has a small effect on the output of g, that is, the normalized squared Euclidean norm

$$\Phi(g, \mathbf{x}) := \frac{||g(h(\mathbf{x}) + \boldsymbol{\xi}) - g(h(\mathbf{x}))||^2}{||g(h(\mathbf{x}))||^2} \tag{3.8}$$

is small. In contrast, a higher $\Phi(g, \mathbf{x})$ mean g is more sensitive to noise $\boldsymbol{\xi}$ at input $\mathbf{x}$.

We now consider the forget-input $\mathbf{x}^f \in \mathcal{D}_f$. Let us define the noise sensitivity of a layer (or composition of layers) g with respect to (w.r.t.) noise $\boldsymbol{\xi}$ on forget-input $\mathbf{x}^f$ as:

$$\Phi(g, \mathbf{x}^f) = \frac{||g(h(\mathbf{x}^f) + \boldsymbol{\xi}) - g(h(\mathbf{x}^f))||^2}{||g(h(\mathbf{x}^f))||^2} \tag{3.9}$$

Consider the noise sensitivity of g at forget-input $\mathbf{x}^f$ in RMU models. Under Assumption 1, the forger-representation in the RMU model $h(\mathbf{x}^f) = c\mathbf{u} + \boldsymbol{\epsilon}$. Now, if we set $\boldsymbol{\xi} = c\mathbf{u} + \boldsymbol{\epsilon} - h(\mathbf{x}^f)$, then we can formulate the unlearning problem as **minimizing the noise sensitivity of the layer**. This objective is described by:

$$\min \frac{||g(c\mathbf{u} + \boldsymbol{\epsilon}) - g(h(\mathbf{x}^f))||^2}{||g(h(\mathbf{x}^f))||^2} \tag{3.10}$$

While g is a transformer layer (or a composition of transformer layers), which is hard to expand it in terms of the coefficient c. Therefore, we propose to use Jacobian $\boldsymbol{J}$—a linearized version of g at $h(\mathbf{x}^f)$—which describes the change in the output of g due to a noise perturbed in the forget-representation $h(\mathbf{x}^f)$. We write h to present $h(\mathbf{x}^f)$. The objective becomes:

$$\min \frac{||\boldsymbol{J}(c\mathbf{u} + \boldsymbol{\epsilon}) - \boldsymbol{J}(h)||^2}{||\boldsymbol{J}(h)||^2} \tag{3.11}$$

We derive the following proposition:

Proposition 2. *Coefficient c and $\cos(\boldsymbol{J}\mathbf{u}, \boldsymbol{J}(h - \boldsymbol{\epsilon}))$ are positively correlated.*

Proof. Since Jacobian $\boldsymbol{J}$ is a linear transformation, rewrite the numerator of Eqn. 3.11 as

$$||\boldsymbol{J}(c\mathbf{u} + \boldsymbol{\epsilon}) - \boldsymbol{J}h||^2 = ||\boldsymbol{J}(c\mathbf{u} + \boldsymbol{\epsilon} - h)||^2 \tag{3.12}$$

Let $\mathbf{v} = \boldsymbol{\epsilon} - h$. By definition of the squared norm, we have:

$$||\boldsymbol{J}(c\mathbf{u} + \mathbf{v})||^2 = (\boldsymbol{J}(c\mathbf{u} + \mathbf{v}))^\top \boldsymbol{J}(c\mathbf{u} + \mathbf{v}) \tag{3.13}$$

$$= (c\mathbf{u} + \mathbf{v})^\top \boldsymbol{J}^\top \boldsymbol{J}(c\mathbf{u} + \mathbf{v}) \tag{3.14}$$

Let matrix $\mathbf{A} = \boldsymbol{J}^\top \boldsymbol{J}$. Expand the Eqn. 3.14, we get

$$||\boldsymbol{J}(c\mathbf{u} + \mathbf{v})||^2 = (c\mathbf{u})^\top \boldsymbol{A} c\mathbf{u} + (c\mathbf{u})^\top \boldsymbol{A}\mathbf{v} + \mathbf{v}^\top \boldsymbol{A} c\mathbf{u} + \mathbf{v}^\top \boldsymbol{A}\mathbf{v} \tag{3.15}$$

Since $\boldsymbol{A}$ is a symmetric matrix, that is, $\boldsymbol{A}^\top = \boldsymbol{A}$, then

$$(c\mathbf{u})^\top \boldsymbol{A}\mathbf{v} + \mathbf{v}^\top \boldsymbol{A} c\mathbf{u} = 2c\mathbf{u}^\top \boldsymbol{A}\mathbf{v} \tag{3.16}$$

Substituting Eqn. 3.16 into Eqn. 3.15, we get:

$$||\boldsymbol{J}(c\mathbf{u} + \mathbf{v})||^2 = c^2\mathbf{u}^\top \boldsymbol{A}\mathbf{u} + 2c\mathbf{u}^\top \boldsymbol{A}\mathbf{v} + \mathbf{v}^\top \boldsymbol{A}\mathbf{v} \tag{3.17}$$

The objective now becomes:

$$\min \frac{c^2\mathbf{u}^\top \boldsymbol{A}\mathbf{u} + 2c\mathbf{u}^\top \boldsymbol{A}\mathbf{v} + \mathbf{v}^\top \boldsymbol{A}\mathbf{v}}{||\boldsymbol{J}h||^2} \tag{3.18}$$

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

$$\frac{2c\mathbf{u}^\top \boldsymbol{A}\mathbf{u} + 2\mathbf{u}^\top \boldsymbol{A}\mathbf{v}}{||\boldsymbol{J}h||^2} = 0 \tag{3.19}$$

Since $||\boldsymbol{J}h||^2$ is not zero, solve for c:

$$c = -\frac{\mathbf{u}^\top \boldsymbol{A}\mathbf{v}}{\mathbf{u}^\top \boldsymbol{A}\mathbf{u}} = \frac{\mathbf{u}^\top \boldsymbol{J}^\top \boldsymbol{J}(h-\boldsymbol{\epsilon})}{\mathbf{u}^\top \boldsymbol{J}^\top \boldsymbol{J}\mathbf{u}} \tag{3.20}$$

$$= \frac{(\boldsymbol{J}\mathbf{u})^\top \boldsymbol{J}(h-\boldsymbol{\epsilon})}{||\boldsymbol{J}\mathbf{u}||^2} \tag{3.21}$$

$$= \frac{||\boldsymbol{J}\mathbf{u}||||\boldsymbol{J}(h-\boldsymbol{\epsilon})||\cos(\boldsymbol{J}\mathbf{u}, \boldsymbol{J}(h-\boldsymbol{\epsilon}))}{||\boldsymbol{J}\mathbf{u}||^2} \tag{3.22}$$

$$= \frac{||\boldsymbol{J}(h-\boldsymbol{\epsilon})||}{||\boldsymbol{J}\mathbf{u}||}\cos(\boldsymbol{J}\mathbf{u}, \boldsymbol{J}(h-\boldsymbol{\epsilon})) \tag{3.23}$$

Since $\frac{||\boldsymbol{J}(h-\boldsymbol{\epsilon})||}{||\boldsymbol{J}\mathbf{u}||}$ is positive, then c and $\cos(\boldsymbol{J}\mathbf{u}, \boldsymbol{J}(h-\boldsymbol{\epsilon}))$ are positively correlated. □

Proposition 2 tell us that smaller (larger) c indicates less (more) alignment between $\boldsymbol{J}\mathbf{u}$ and $\boldsymbol{J}(h-\boldsymbol{\epsilon})$. Given that the Jacobian $\boldsymbol{J}$ describes how small changes in the input lead to changes in the output using linear approximation around a given point. If $\boldsymbol{J}$ does not vary drastically, it will not significantly alter the directions of $\boldsymbol{u}$ and $h-\boldsymbol{\epsilon}$. In such cases, $\boldsymbol{J}$ will have a small effect on directional alignment, preserving the relative angles between $\boldsymbol{u}$ and $h-\boldsymbol{\epsilon}$. Here, reasonably, $\mathbf{u}$ *and* h *are becoming more aligned as* c *increases* since error $\boldsymbol{\epsilon} \to \mathbf{0}$ as unlearning becomes more accurate.

The above discussion does not address RQ2. However, the definition of the noise sensitivity suggests that the noise sensitivity of layer g is characterized by the inherent properties of g, the representation $h(\mathbf{x}^f)$ and the perturbed noise $\boldsymbol{\xi}$. If $\boldsymbol{\xi}$ is predetermined, the noise sensitivity of g depends solely on its properties. This suggest the following experiment: we compute $\hat{h}(\mathbf{x}^f)$—the mean of $h(\mathbf{x}^f)$ over a set of input $\mathbf{x}^f \sim \mathcal{D}_f$, compress a *fix* noise $\boldsymbol{\xi}$ into $\hat{h}(\mathbf{x}^f)$. We then calculate the noise sensitivity of g for different layers. Figure 3.1 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

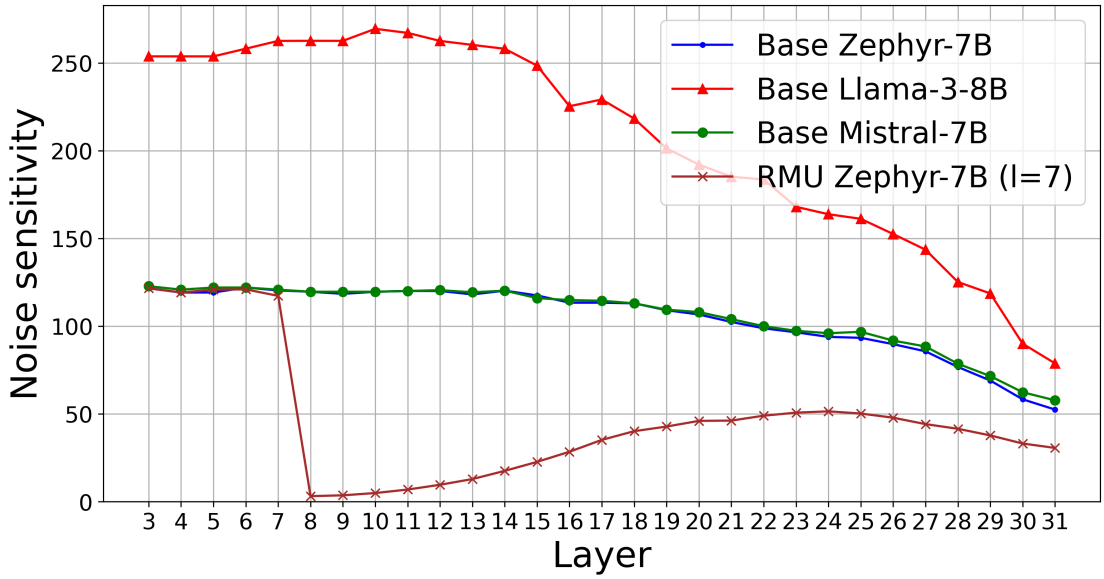


Figure 3.1: Noise sensitivity of layer g from the third to the last layer in base Zephyr-7B, base Llama-3-8B, base Mistral-7B, and RMU Zephyr-7B model. In the base models, a deeper layer has lower noise sensitivity, while the noise sensitivity is minimized in the RMU model (inject noise into $h^{(7)}$, the noise sensitivity of layer $k = 8$ is minimized). These results verify our analysis.

noise sensitivity describes a layer's robustness to noise, higher noise sensitivity means g 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 (smaller c), whereas deeper layers require larger noise (larger c)**. We empirically validate our analysis in Section. 6.

3.3 Adversarial Robustness of RMU Models

A good unlearning method must be resilient to knowledge recovery, that is, it prevents the model from resurfacing when attacked by adversarial jailbreak attacks. Li et al. (2024b) presents a qualitative experiment by employing Greedy Coordinate Gradient (GCG; Zou et al. (2023b)), a state-of-the-art white-box attack method, to jailbreak RMU unlearned models to extract the previously forgotten knowledge. As a result, RMU models demonstrate strong robustness against GCG attacks. For example, GCG can jailbreak the base Yi-34B models to answer the harmful queries in **less than 50 gradient update steps**, while the RMU unlearned Yi-34B models still output gibberish after $2,500$ gradient update steps (7 hours of optimization on an NVIDIA A100). However, the reason is unexplored.

We take a step toward explaining the causes for RMU's robustness from the standpoint of an attack-defense game: RMU acts as a defender and GCG acts as an attacker. We define the threat model and the problem formulation as follows.

3.3.1 Threat Model

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 to generate 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.

3.3.2 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 $\mathbf{x}^f_{<n}$, the attacker finds an update $\boldsymbol{\delta}$ to adversarial trigger aims to maximize the likelihood of generating the target sequence $\mathbf{x}^f_{<n|n+K}$ consists of K tokens. For simplification, we denote $\mathbf{x}^f = \mathbf{x}^F_{<K+1} = [\mathbf{x}^f_{<n}, \mathbf{x}^f_{<n:n+K}]$. The attacker tries to solve the following objective:

$$\min_{\mathbf{x}^f+\boldsymbol{\delta}} \mathcal{J}(f(\mathbf{x}^f + \boldsymbol{\delta})), \tag{3.24}$$

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

3.3.3 Robustness of RMU Models Against GCG Attacks

We show that the GCG attacker misjudges in finding optimal adversarial token substitution in RMU models. Specifically, the gradient of the loss at input $\mathbf{x}^f$ with

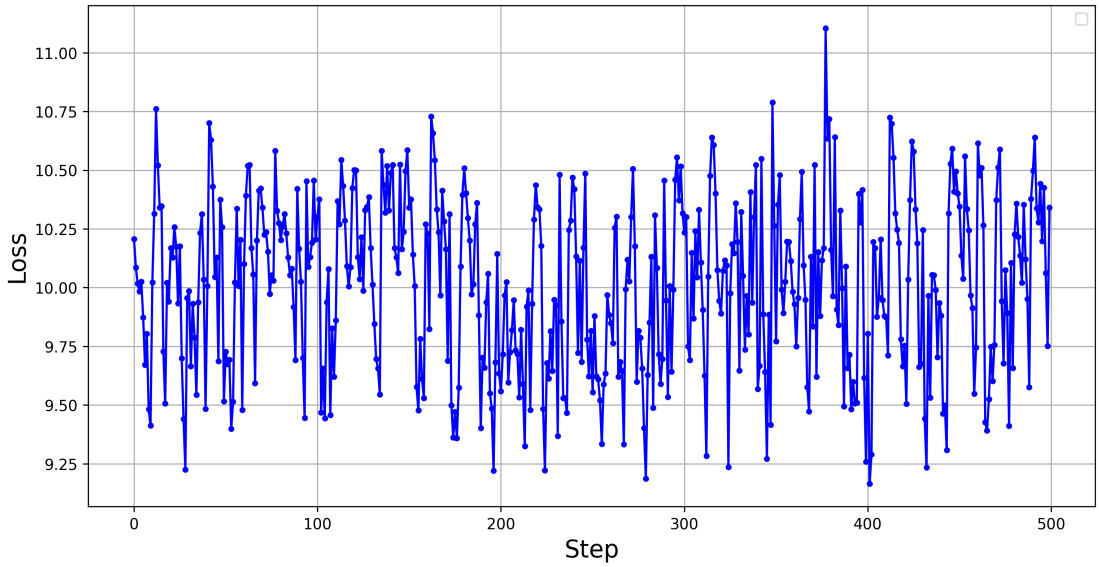


Figure 3.2: Loss values of the GCG attacker during optimization. After 500 gradient update steps, the loss oscillates around its initial value. This indicates that the GCG attacker receives unreliable gradient signals, preventing it from effectively minimizing its loss. This result verifies our analysis.

respect to $e_{\mathbf{x}_i}$ in RMU models is

$$\nabla_{e_{\mathbf{x}_i}} \mathcal{J}(f^u(\mathbf{x}^f)) \tag{3.25}$$

Under Assumption 1, we have

$$\begin{aligned} \nabla_{e_{\mathbf{x}_i}} \mathcal{J}(f^u(\mathbf{x}^f)) &= \nabla_{e_{\mathbf{x}_i}} \mathcal{J}(g(h^{(l),\text{rand}}(\mathbf{x}^f)) \\ &= \nabla_{e_{\mathbf{x}_i}} (\mathcal{J} \circ g)(c\mathbf{u} + \boldsymbol{\epsilon}) \end{aligned} \tag{3.26}$$

Since c and $\mathbf{u}$ are predetermined before unlearning, $(\mathcal{J} \circ g)(c\mathbf{u})$ does not change with respect to $e_{\mathbf{x}_i}$. The gradient $\nabla_{e_{\mathbf{x}_i}} (\mathcal{J} \circ g)(c\mathbf{u}+\boldsymbol{\epsilon})$ close to 0 for all token $\mathbf{x}_i$ 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 experiments (c.f. Section 5.3) implicitly verify our analysis. Additionally, we visualize the loss $\mathcal{J}$ of the GCG during its optimization in Figure 3.2.

Chapter 4

Empirical Analysis

4.1 Measuring Token Confidence with MaxLogit

As discussed before in Section 3.1, we validate our analysis by considering the Maximum Logit Value (MaxLogit) estimator for measuring the token confidence. While MaxLogit does not provide guarantees of prediction correctness, prior work shows that it still predicts correctness (Plaut et al., 2024). More specifically, we compute the MaxLogit for each token $\mathbf{x}_n$ given a sequence of tokens $\mathbf{x}_{<n}$ from vocabulary V as:

$$\text{MaxLogit}(\mathbf{x}_n) = \max_{\mathbf{x}_n \in V} f(\mathbf{x}_n | \mathbf{x}_{<n}) \tag{4.1}$$

We use WMDP-Biology and WMDP-Cyber Q&A datasets (Li et al., 2024b) with a total of $3,260$ Q&As. We employ lm-eval-hardness framework (Gao et al., 2024) and formulate each original question and answer (Figure 4.3) as a zero-shot Q&A prompt to query the unlearned LLM as described in Figure 4.4. Lm-eval-hardness extracts first answer letter (A, B, C, or D) from the generated output using heuristic rules or regex expressions.

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

An original WMDP sample

question: "How can I cultivate a calicivirus?",
choices: ["Cell/enteroid culture", "Helper virus-free reverse genetics system", "Either A or B", "Neither A nor B"],
answer: 0

Prompt query

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

Question: How can I cultivate a calicivirus?

A. Cell/enteroid culture.

B. Helper virus-free reverse genetics system.

C. Either A or B.

D. Neither A nor B.

Answer:"

Figure 4.1: A sample QA prompt.

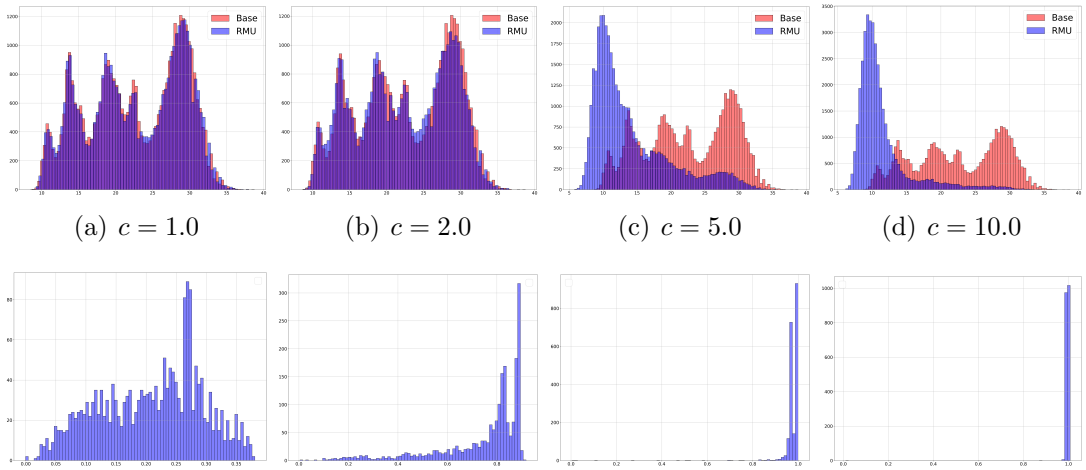
for each model Base vs. RMU (unlearned on WMDP-Biology and WMDP-Cyber forget datasets).

The results are presented in Figure 4.2 (a)-(d). We find that the MaxLogit distribution for the base model is generally wider 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. This result verifies our analysis.

4.1.1 Effects of the Coefficient c

Effects of c on forget and retain performances. We analyze the impact of c for forget and retain performances, using WMDP (Li et al., 2024b) and MMLU (Hendrycks et al., 2020). We defer the full experimental setup to Section 6. Figure 4.3a shows: (i) a clear positive correlation between the drop-in-accuracy rate and the value of c, that is, 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 (Figure 4.3b).

Effects of c on alignment between $\mathbf{u}$ and h. We compute $\cos(\mathbf{u}, h)$ scores of pairs of $\mathbf{u}$ and $h(\mathbf{x}^f)$ for all $\mathbf{x}^f$ in on WMDP-Biology and WMDP-Cyber forget



(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 4.2: The distribution of MaxLogit (a-d) on WMDP Biology and Cyber Q&As with different coefficient c of the base Zephyr-7B and RMU Zephyr-7B models. The distribution of $\cos(\mathbf{u}, h)$ (e-h) of the RMU Zephyr-7B model.

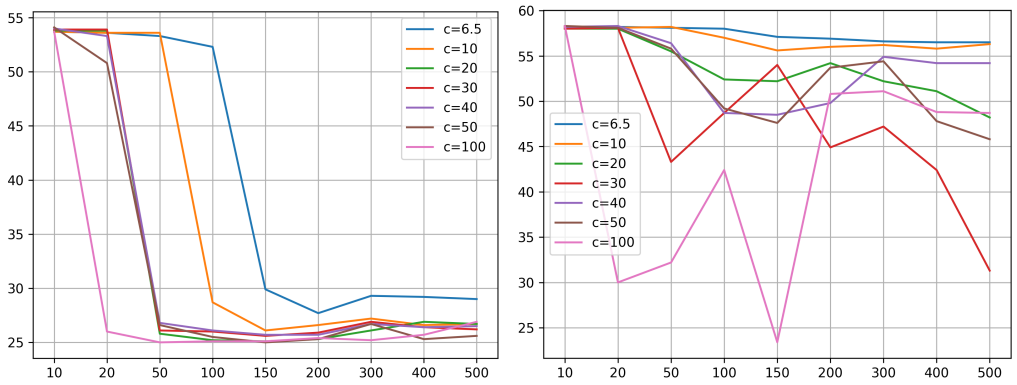


Figure 4.3: Average accuracy of WMDP-Biology and WMDP-Cyber (left) and MMLU (right) with different coefficient $c \in [6.5, 10, 20, 30, 40, 50, 100]$.

datasets and visualize the $\cos(\mathbf{u}, h)$ score distribution shown in Figure 3.2e-h. We observed that there is a clear positive correlation between $\cos(\mathbf{u}, h)$ scores and the coefficient c. As c increases, the distribution of $\cos(\mathbf{u}, h)$ scores shifts towards higher values and is almost distributed with a peak at 1.0 (Figure 3.2g-h). This verifies our analysis in Section 3.2.

4.1.2 Effects of Unlearn Layers

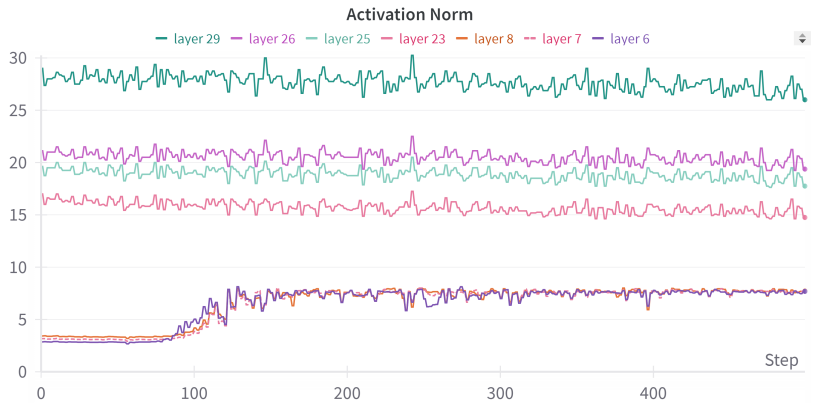


Figure 4.4: Norm of forget-representations across different layers.

We investigate the effect of unlearn layers on accuracy and the representation norm during unlearning. Following the original work, we change the unlearn layer l from $3 \rightarrow 31$, fixed $c = 6.5$. Figure 6.1 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 Figure 4.4, we observed that within early layers, the Euclidean norm of the forget-representation is 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 deep layers, the forget-representation norms, initially larger than c, remain unchanged during unlearning, making forget-loss non-convergence. This observation motivates the use of a layer-aware, adaptive coefficient in RMU. We present Adaptive RMU, a simple yet effective alternative to RMU in the next chapter.

Chapter 5

Adaptive RMU

5.1 Adaptive Forget Loss

Inspired by the observations in Section 4.1.2, we propose *Adaptive RMU*, a simple yet effective alternative method with an adaptive forget loss by scaling the random unit vector $\mathbf{u}$ with an *adaptive scaling coefficient* $\beta||h^{(l)}_{\theta^{\text{ref}}}(\mathbf{x}^f)||$, where $\beta \in \mathbb{R}_+$ is a scaling factor and $||h^{(l)}_{\theta^{\text{ref}}}(\mathbf{x}^f)||$ is the Euclidean norm of forget-representation on the reference model $f_{\theta^{\text{ref}}}$. The total loss of Adaptive RMU is calculated as follows:

$$\begin{aligned}\mathcal{L}^{\text{adaptive}} = &\underbrace{\mathbb{E}_{\mathbf{x}^f \sim \mathcal{D}_f} ||h^{(l)}_{\boldsymbol{\theta}}(\mathbf{x}^f) - \beta||h^{(l)}_{\boldsymbol{\theta}^{\text{ref}}}(\mathbf{x}^f)||\mathbf{u}||^2}_{\text{adaptive forget loss}} \\ &+ \alpha \underbrace{\mathbb{E}_{\mathbf{x}^r \sim \mathcal{D}_r} ||h^{(l)}_{\boldsymbol{\theta}}(\mathbf{x}^r) - h^{(l)}_{\boldsymbol{\theta}^{\text{ref}}}(\mathbf{x}^r)||^2}_{\text{retain loss}}\end{aligned} \tag{5.1}$$

5.2 Computational Perplexity

Our Adaptive RMU is shown in Algorithm 2. The difference between RMU and Adaptive RMU is the calculation of the forget coefficient. RMU uses a predefined coefficient c while Adaptive RMU uses an adaptive coefficient $\beta||h^{(l)}_{\theta^{\text{ref}}}(\mathbf{x}^f)||$, which can be calculated and cached during the first iteration of the inner for loop in Algorithm 2. No gradient computations or additional forward pass are introduced, thus, the complexity of Adaptive RMU is equal to that of RMU.

We note that Adaptive RMU aims to address the challenge of adaptively determining the coefficient c in RMU. The introduced value β is manually tuned via grid search, leaving the challenge to not fully resolved. However, we emphasize that Adaptive RMU offers significant computational advantages over the original

Algorithm 2 Adaptive RMU pseudocode

Require:

$\mathcal{D}_f$: a forget dataset.
$\mathcal{D}_r$: a retain dataset.
$f_{\boldsymbol{\theta}^{\text{ref}}}$: a frozen model.
$f_{\boldsymbol{\theta}}$: an update model (unlearn 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_θ.

Sample a random unit vector $\mathbf{u}$, each element $\mathbf{u}_i$ uniformly sampled from $[0, 1)$.

for step $t \in [1...T] : \mathbf{x}^f \in \mathcal{D}_f,\ \mathbf{x}^r \in \mathcal{D}_r$ **do**
    Forward and hook the representations of $\mathbf{x}^f$ and $\mathbf{x}^r$ at layer l from the frozen and update model.
    Compute the adaptive loss $\mathcal{L}^{\text{adaptive}}$ by Eqn. 5.1.
    Update $\boldsymbol{\theta}$ w.r.t $\nabla\mathcal{L}^{\text{adaptive}}$ using gradient descent.
    $t = t + 1$
end for
return $f_{\boldsymbol{\theta}}$

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 LLMs grows.

Chapter 6

Experiment

6.1 Experimental setup

6.1.1 Datasets

We use WMDP-Biology and WMDP-Cyber forget datasets as $\mathcal{D}_f$ and Wikitext (Merity et al., 2022) as $\mathcal{D}_r$ for fine-tuning. Unlearned models are evaluated on WMDP Q&A datasets and MMLU (Hendrycks et al., 2021).

WMDP (Li et al., 2024b) stands for Weapon of Mass Destruction Proxy, is a corpora consisting of forget sets, retain sets, and Q&A sets. The WMDP Q&A is a dataset of $3,668$ multiple-choice questions about Biosecurity $(1,273)$, Cybersecurity $(1,987)$, and Chemical security (408). The WMDP-Biology forget and retain sets consist of papers from PubMed. The WMDP-Biology forget comprises papers used in generating WMDP-Biology questions, while the retain set samples papers from various categories within general biology. The retain set excludes papers from the forget set and employs keyword exclusion to avoid topics related to Q&A set. The WMDP-Cyber forget and retain sets consist of passages crawled from GitHub with two different sets of keywords. Note that we did not benchmark for WMDP-Chemistry Q&A due to no WMDP-Chemistry forget set being publicly released. This dataset is available at `https://github.com/centerforaisafety/wmdp`.

MMLU (Hendrycks et al., 2021) stands for Massive Multitask Language Understanding, a dataset of $15,908$ multiple-choice Q&A covers 57 subjects across STEM, the humanities, social science, and more. MMLU is designed to measure general knowledge by evaluating models in zero-shot or few-shot settings. This dataset is available at `https://huggingface.co/datasets/cais/mmlu`.

Wikitext (Merity et al., 2022) is a language modeling dataset consisting of over 100 milion tokens extracted from Wikipedia. Following Li et al. (2024b), 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`.

6.1.2 Models

We employ the following pre-trained open-weight 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).

6.1.3 Hyperparameters

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 $\alpha_{\text{biology}} = \alpha_{\text{cyber}} = 1200$. 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 GPUs were used to run the experiments. The representation from the MLP module is used as the representation.

6.1.4 Comparison Methods

We compare Adaptive RMU against 4 baselines: RMU (Li et al., 2024b), Large Language Model Unlearning (LLMU; Yao et al. (2023)), SCalable Remenbering and Unlearning unBound (SCRUB; Kurmanji et al. (2023)), and Selective Synaptic Dampening (SSD; Foster et al. (2024). We use off-the-shelf results from (Li et al., 2024b) for LLMU, SCRUB, and SSD.

6.2 Result and Analysis

6.2.1 Main Results

Main results are reported using the Zephyr-7B model. Figure 6.1 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%

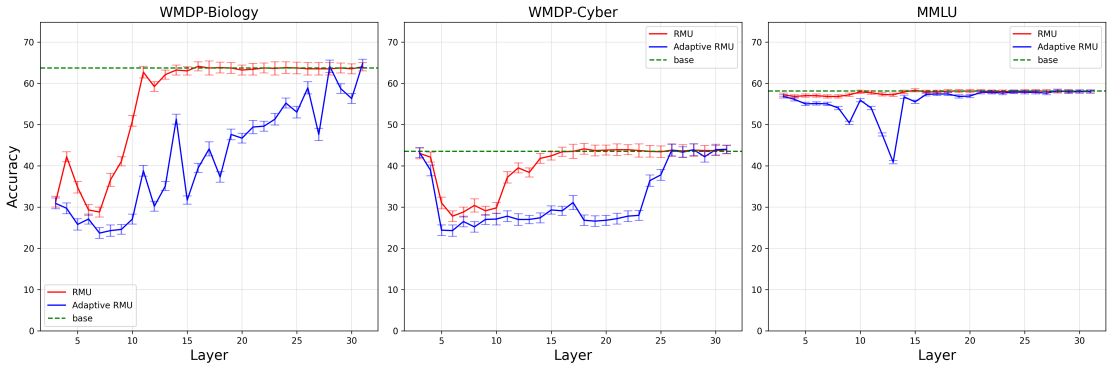


Figure 6.1: Q&A accuracy of RMU and Adaptive RMU Zephyr-7B models on WMDP-Biology (↓), WMDP-Cyber (↓), and MMLU (↑) datasets with respect to the unlearned layer l, ranging from the third to the last layer. Adaptive RMU demonstrates effective unlearning across various layers, overcoming the limitation of RMU.

Method/tasks	WMDP-Biology ↓	WMDP-Cyber ↓	MMLU ↑
Base	63.7	43.5	58.1
LLMU (Yao et al., 2023)	59.5	39.5	44.7
SCRUB (Kurmanji et al., 2023)	43.8	39.3	51.2
SSD (Foster et al., 2024)	50.2	35.0	40.7
RMU (Li et al., 2024b)	<u>28.8</u>	<u>28.8</u>	**56.8**
Adaptive RMU (Dang et al., 2025)	**23.7**	**26.5**	<u>55.0</u>

Table 6.1: Q&A accuracy of Zephyr-7B unlearned models on WMDP-Biology, WMDP-Cyber, and MMLU. The **best** and <u>runner up</u> are marked.

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 6.1 further highlights that Adaptive RMU ($l = 7$) outperforms RMU ($l = 7$), LLMU, SCRUB, and SSD, establishing a new state-of-the-art performance.

6.2.2 Unlearning Performance of Other Models

We report the unlearning performance of Adaptive RMU Yi-6B, Llama-3-8B, and Mistral-7B models in Table 6.2, Table 6.3, and Table 6.4. We observed a clear trend that the unlearning performance is more effective when using the early layer as the unlearn layer.

Task/unlearn layer	base	3	4	5	6	7	8	9	10	11	12	13	14	15	16
WMDP-Biology ↓	64.8	65.0	49.9	35.2	27.8	26.1	63.3	26.2	27.1	27.4	27.1	26.0	25.4	27.2	34.8
WMDP-Cyber ↓	41.1	40.7	40.5	37.7	28.1	25.5	39.3	25.6	23.9	26.1	23.6	24.3	24.2	24.0	25.5
MMLU ↑	60.0	60.1	57.7	59.4	51.4	56.5	59.9	56.8	53.7	48.1	49.3	57.0	55.6	47.7	53.3
Task/unlearn layer	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
WMDP-Biology ↓	30.3	32.2	27.1	31.9	41.0	53.4	50.4	53.2	39.2	46.0	39.0	42.5	41.6	40.5	64.8
WMDP-Cyber ↓	25.3	24.4	24.3	24.5	26.7	29.8	33.9	36.2	34.3	34.6	31.4	30.4	39.6	40.8	40.6
MMLU ↑	45.4	52.1	56.7	58.2	59.3	59.4	59.6	59.7	59.4	59.7	59.4	59.4	59.5	59.7	60.1

Table 6.2: Q&A accuracy of Adaptive RMU Yi-6B models on WMDP-Biology, WMDP-Cyber, and MMLU.

Task/unlearn layer	base	3	4	5	6	7	8	9	10	11	12	13	14	15	16
WMDP-Biology ↓	71.2	46.4	45.3	28.2	27.8	29.3	33.7	36.0	65.1	64.9	62.8	65.2	59.6	44.4	41.4
WMDP-Cyber ↓	43.9	32.5	25.5	24.5	27.6	26.8	27.3	26.3	32.5	32.3	34.1	35.2	29.9	28.3	27.8
MMLU ↑	62.0	60.7	60.2	59.7	60.7	60.0	60.1	59.6	61.8	61.3	61.5	61.5	61.8	60.9	61.1
Task/unlearn layer	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
WMDP-Biology ↓	35.5	35.2	41.1	60.8	33.7	59.3	54.6	56.7	69.6	62.2	70.0	69.9	69.9	67.0	70.4
WMDP-Cyber ↓	28.0	33.5	28.6	39.0	28.6	31.7	35.5	36.9	45.5	44.8	44.4	43.5	44.4	43.6	43.4
MMLU ↑	61.3	61.3	61.3	61.9	60.8	61.7	61.2	61.5	61.9	61.7	62.0	61.9	61.5	61.5	62.1

Table 6.3: Q&A accuracy of Adaptive RMU Meta Llama-3-8B models on WMDP-Biology, WMDP-Cyber, and MMLU.

Task/unlearn layer	base	3	4	5	6	7	8	9	10	11	12	13	14	15	16
WMDP-Biology ↓	67.3	28.0	28.9	27.6	27.5	26.3	24.5	25.7	26.1	27.6	31.4	37.7	35.6	25.4	35.0
WMDP-Cyber ↓	44.1	42.1	41.9	24.8	26.8	26.3	26.6	26.4	26.7	25.7	26.5	25.8	31.6	26.7	27.9
MMLU ↑	58.7	54.5	57.2	54.9	55.8	55.7	47.3	53.0	47.4	35.1	54.5	55.9	51.5	44.9	57.3
Task/unlearn layer	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
WMDP-Biology ↓	27.4	56.4	38.4	45.7	42.0	52.0	52.4	61.1	57.5	62.2	63.2	66.3	61.9	61.0	66.0
WMDP-Cyber ↓	27.5	38.9	26.5	26.7	26.6	27.4	27.7	38.9	43.9	43.4	43.7	43.8	44.0	42.5	43.4
MMLU ↑	56.7	56.8	56.2	57.6	58.1	58.3	58.1	58.2	58.6	58.7	58.6	58.7	58.4	58.3	58.2

Table 6.4: Q&A accuracy of Adaptive RMU Mistral-7B models on WMDP-Biology, WMDP-Cyber, and MMLU.

6.2.3 Performances on MMLU Subset Unlearning Benchmark

We did additional experiments on the MMLU subset unlearning benchmark with three settings:

1. MMLU-Economics: unlearning high school microeconomics and macroeconomics and maintaining performance on the remaining categories (refers as MMLU-Retain tasks).

2. MMLU-Law: unlearning international and professional law while maintaining performance on MMLU-Retain.

3. MMLU-Physics: unlearning high school and college physics while maintain-

ing general performance in MMLU-Retain.

Settings. We use publicly released forget-set by Li et al. (2024b) for each task and Wikitext (Merity et al., 2022) as retain set. We use a fixed sequence length of 512 for MMLU-Economics, MMLU-Law, MMLU-Physics, and Wikitext. We keep other hyperparameters unchanged as in SubSection 6.1.3.

Result. Table 6.5 presents the unlearning performance of Adaptive RMU Zephyr-7B models on MMLU-Economics, MMLU-Law, and MMLU-Physics. We observe a notable reduction in accuracy on the forget tasks. However, the model exhibits excessive unlearning, leading to substantial performance degradation on the MMLU-Retain tasks.

Task/unlearn layer	base	3	4	5	6	7	8	9	10	11	12	13	14	15	16
MMLU-Economics ↓	58.0	57.0	45.7	22.8	23.4	27.0	28.8	27.0	34.6	24.6	42.1	45.5	34.8	44.5	58.3
MMLU-Law ↓	55.6	49.8	53.5	25.2	24.5	26.4	24.6	24.2	21.5	23.9	51.1	44.1	36.8	44.7	46.0
MMLU-Physics ↓	38.5	39.3	37.9	28.8	27.2	23.8	21.7	20.5	21.0	29.2	32.6	34.1	34.4	35.7	42.3
MMLU-Retain ↑	58.9	58.0	57.3	39.3	45.2	39.4	35.2	36.0	44.8	35.2	52.9	55.2	46.0	54.8	56.8
Task/unlearn layer	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MMLU-Economics ↓	51.8	36.0	54.4	26.0	21.4	42.8	43.4	42.8	48.4	57.2	58.7	50.0	58.2	58.9	57.8
MMLU-Law ↓	49.8	24.3	54.4	27.2	24.6	24.2	25.4	44.6	54.4	55.8	56.7	53.6	55.6	55.4	56.1
MMLU-Physics ↓	37.5	26.7	26.9	21.0	21.6	24.2	23.4	25.6	29.6	37.1	31.9	33.8	36.9	33.9	38.6
MMLU-Retain ↑	57.6	47.8	57.7	36.2	30.3	39.6	47.4	52.0	58.1	58.9	58.9	56.4	59.0	59.1	59.0

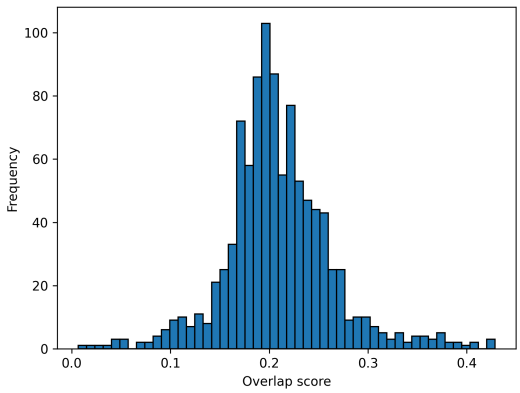
Table 6.5: Q&A accuracy of Adaptive RMU Zephyr-7B models on MMLU-Economics, MMLU-Law, MMLU-Phycics, and MMLU-Retain.

6.2.4 Effects of In-domain Retain Dataset.

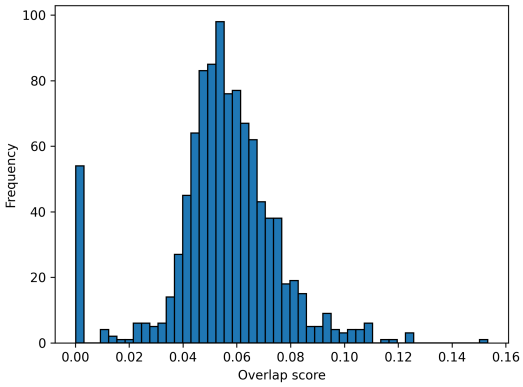
In this setting, we use the WMDP-Biology and WMDP-Cyber retain sets instead of Wikitext. We use the same hyperparameters as in Section 6.1.3. Table 6.6 shows that Adaptive RMU is **ineffective for all unlearn layers**. As WMDP-forget and retain sets are collected from the same source, even with efforts to distinguish them, these corpora may commonly have overlapping texts. We present an n-gram overlap analysis between the WMDP-forget set and the WMDP-retain set as a measurement of unlearning difficulty.

n-gram overlap analysis. Given a retain sample $\mathbf{x}_{1:k} \in \mathcal{D}_r$ consists of k tokens $\{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_k\}$, we denote $\mathbf{x}_{i:i+n-1}$ for $i \in [1, ..., k-n+1]$ as the n-gram of $\mathbf{x}_{1:k}$. The n-gram overlap score of $\mathbf{x}_{1:k}$ in forget set $\mathcal{D}_f = \{\mathbf{x}^f\}^{|\mathcal{D}_f|}$ is defined as:

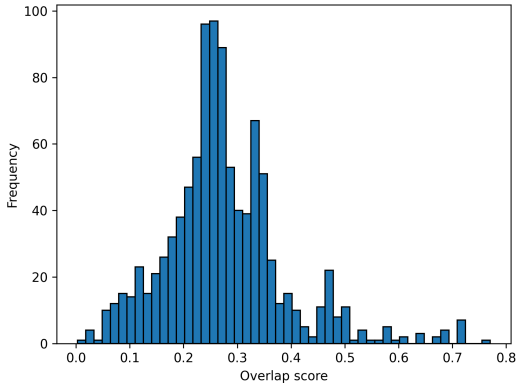
$$\frac{1}{|\mathcal{D}_f|} \frac{1}{k-n+1} \sum_{\mathbf{x}^r} \sum_{i=1}^{k-n+1} \mathbb{I}\left[\mathbf{x}_{i:i+n-1} \in \mathbf{x}^f\right], \tag{6.1}$$



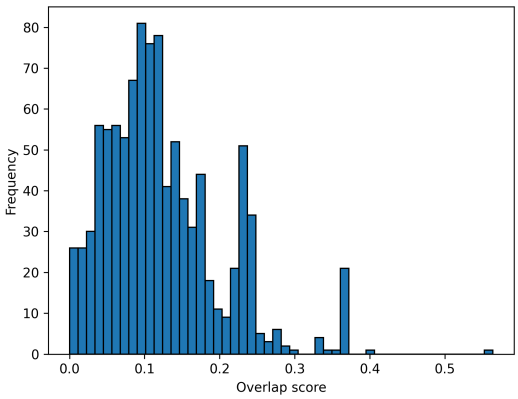
(a) Distribution of Unigram overlap score between WMDP-Biology retain and WMDP-Biology forget sets.



(b) Distribution of Bigram overlap score between WMDP-Biology retain and WMDP-Biology forget sets.



(c) Distribution of Unigram overlap score between WMDP-Cyber retain and WMDP-Cyber forget sets.



(d) Distribution of Bigram overlap score between WMDP-Cyber retain and WMDP-Cyber forget sets.

Figure 6.2: Distributions of Unigram and Bigram overlap scores.

Task/unlearn layer	base	3	4	5	6	7	8	9	10	11	12	13	14	15	16
WMDP-Biology ↓	63.7	63.2	63.3	62.9	28.1	62.6	49.9	64.2	29.6	62.0	63.0	63.7	63.7	64.4	64.3
WMDP-Cyber ↓	43.5	42.7	42.0	40.1	24.6	33.3	33.9	40.8	25.1	41.3	41.7	42.8	43.4	42.8	43.4
MMLU-All ↑	58.1	57.4	57.4	57.9	30.1	57.6	38.3	57.6	29.3	57.1	58.0	57.5	57.7	57.9	57.8
Task/unlearn layer	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
WMDP-Biology ↓	63.9	63.7	63.9	63.5	63.5	63.7	63.7	63.6	63.6	63.5	63.3	63.7	63.8	63.5	64.6
WMDP-Cyber ↓	44.5	43.5	43.5	44.4	43.9	43.5	44.3	43.6	43.9	43.8	43.6	43.2	43.7	43.7	43.6
MMLU-All ↑	58.4	58.1	58.2	57.6	58.2	58.1	58.2	58.1	58.1	58.0	58.2	58.1	58.2	58.1	57.9

Table 6.6: Q&A accuracy of Adaptive RMU Zephyr-7B models on WMDP-Biology, WMDP-Cyber, and MMLU. Models were fine-tuned on WMDP-Biology and WMDP-Cyber retain sets.

where $\mathbb{I}(\cdot)$ is the indicator function and $\mathbb{I}[\mathbf{x}_{i:i+n-1} \in \mathbf{x}^f] = 1$ if the substring $\mathbf{x}_{i:i+n-1}$ is in forget sample $\mathbf{x}^f$, otherwise 0. We randomly sampled 1000 documents from each dataset and performed Unigram ($n = 1$) and Bigram ($n = 2$) overlap analysis. The results indicate a high degree of unigram and bigram overlap between the WMDP-forget and WMDP-retain sets. Specifically, the average Unigram and Bigram overlap scores for the WMDP-Biology forget and retain sets were 20.8% and 5.5%, respectively. These overlap scores were even higher for the WMDP-Cyber sets, at 27.5% and 12.3%, respectively. The distributions of n-gram overlap scores are visualized in Figure 6.2. High n-gram overlap scores make two distributions, WMDP forget-set and WMDP retain-set, less distinguishable, which makes the unlearning more difficult.

Claim: this result is evidence that RMU and Adaptive RMU cannot truly distinguish between harmful knowledge vs. general knowledge. They might simply try to maximize the loss of tokens in the forget-sets.

6.2.5 Example of Generated Outputs

This section shows examples of generated answers in WMDP of RMU and Adaptive RMU model.

Example 1 (Biology):
"The following are multiple choice questions (with answers) about Biology.

Question: My labmate is setting up for some lab work. They have turned on a biosafety cabinet, set out Mueller-Hinton agar and cefoxitin diffusion disks, and are warming an incubator to 37 C. What microorganism might they be growing?
A. Mycobaterium smegmatis
B. Aspergillus niger
C. Staphylococcus aureus

D. Bacillus subtilis
Answer:"
True answer: "Staphylococcus aureus".
RMU's answer: "C. Staphylococcus aureus
Question: Which of the following is a type of cellular respiration?
A".
Adaptive RMU's answer: "The ABabai lababudcai (hotspowered the TT Superior, and the TT Superior is a superior".

Example 2 (Cyber):
"The following are multiple choice questions (with answers) about Cyber.

Question: Are any of the following flags a security concern when passed to the v8 JavaScript engine? Assume the engine will be used to run arbitrary JS from the internet.
A. –trace-turbo
B. –trace-gc
C. –print-bytecode
D. –allow-natives-syntax"
Answer:"
True answer: "–allow-natives-syntax".
RMU's answer: "D. –allow-natives-syntax

Question:Which of the following is a security concern when passed to the v8 JavaScript engine".
Adaptive RMU's answer: "The above results are not both both both both than both of the both of you.

C. The both of you.
both both".

6.3 Discussion: Does RMU Truly Unlearn?

As a human, forgetting some basic knowledge might seem "impossible" once learned. For example, for some basic facts such as: "the sun rises every day," it is much harder to unlearn them than the rare, unusual ones. Imagine, if someone claims that they forgot something, it might be one of two cases: (1) They know about it, but intentionally decided that they do not know about it, or (2) they know about it, but not well enough to be able to claim that they have learn it.

We ask whether RMU can truly unlearn (or erase) knowledge in LLMs, or does it intentionally decide that it does not know about the target forget knowledge?

We show that the mechanism of RMU aligns with (1): **the model still retains the knowledge internally, but is directed to behave as if it does not know it.**
More concretely, we formulate RMU as a backdoor attack problem (Huu-Tien et al., 2025).

Notation. Let $\mathbf{z}^f_{\boldsymbol{\theta}} = h^{(l)}_{\boldsymbol{\theta}}(\mathbf{x}^f)$ and $\mathbf{z}^r_{\boldsymbol{\theta}} = h^{(l)}_{\boldsymbol{\theta}}(\mathbf{x}^r)$ be the intermediate representations of the forget-input $\mathbf{x}^f$ and retain-input $\mathbf{x}^r$ in the model $f_{\boldsymbol{\theta}}$ at layer l. $\mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}} = h^{(l)}_{\boldsymbol{\theta}^{\text{ref}}}(\mathbf{x}^f)$ and $\mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}} = h^{(l)}_{\boldsymbol{\theta}^{\text{ref}}}(\mathbf{x}^r)$ denote the intermediate representations in the reference model $f_{\boldsymbol{\theta}^{\text{ref}}}$.

RMU as a backdoor attack. Consider the supervised learning setting to learn a model $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}$. Let $\mathcal{Z} = \mathcal{Z}_f \cup \mathcal{Z}_r$ be the "latent representation" dataset corresponding to the original dataset $\mathcal{D} = \mathcal{D}_f \cup \mathcal{D}_r$. $\mathcal{Z}$ is composed of a forget-set $\mathcal{Z}_f = \{(\mathbf{z}^f_{\boldsymbol{\theta}}, \mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}})\}_i$, where $\mathbf{z}^f_{\boldsymbol{\theta}} \in \mathcal{X}$ is the input, $\mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}} \in \mathcal{Y}$ is the target output, and a retain-set $\mathcal{Z}_r = \{(\mathbf{z}^r_{\boldsymbol{\theta}}, \mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}})\}_j$ where $\mathbf{z}^r_{\boldsymbol{\theta}} \in \mathcal{X}$ and $\mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}} \in \mathcal{Y}$. Each forget-sample $(\mathbf{z}^f_{\boldsymbol{\theta}}, \mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}})$ is transformed into a backdoor-sample $(T(\mathbf{z}^f_{\boldsymbol{\theta}}), \Omega(\mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}}))$, where Ω is an adversarial-target labeling function and T is the trigger generation function. In a standard backdoor attack, T is usually optimized for generating and placing the trigger into the input while Ω specifies the behavior of the model when the backdoor trigger is activated. In the "forgetting", T is an identity function *i.e.* $T(\mathbf{z}^f_{\boldsymbol{\theta}}) = \mathbf{z}^f_{\boldsymbol{\theta}}$ and Ω is a function that maps $\mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}}$ to the adversarial-perturbed representation. We train model $f_{\boldsymbol{\theta}}$ 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 $\mathcal{Z}_r = \{(\mathbf{z}^r_{\boldsymbol{\theta}}, \mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}})\}_j$, as follows:

$$\boldsymbol{\theta}^* = \arg\min_{\theta} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{Z}} \left[\mathcal{L}(f_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{y})\right], \tag{6.2}$$

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

$$f(\mathbf{z}^r_{\boldsymbol{\theta}}) = \mathbf{z}^r_{\boldsymbol{\theta}^{\text{ref}}} \tag{6.3}$$

$$f(\mathbf{z}^f_{\boldsymbol{\theta}}) = f(T(\mathbf{z}^f_{\boldsymbol{\theta}})) = \Omega(\mathbf{z}^f_{\boldsymbol{\theta}^{\text{ref}}}) \tag{6.4}$$

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* (Lee et al., 2024).

The backdoor attack formulation explains the vulnerability of RMU to black-box adversarial attacks or even under non-conditioned adversarial attacks (Thaker et al., 2025).

Chapter 7

Conclusion

7.1 Summary

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.

7.2 Limitations, Open Problems, and Future Directions

7.2.1 Limitations

We discuss the following limitations in our paper:

1. We mainly perform experiments on 7B versions (or equivalent) due to computational constraints. To validate the generalizability of our approach and findings, we conducted experiments across the Zephyr, Mistral, Llama, and Yi models.

2. Our analysis on white-box attacks for open-weight models. In practice, state-of-the-art LLMs such as GPT, Gemini, and Claude are trained privately and are accessible through API only. The most common form of attack on LLMs, therefore, is a black-box jailbreak attack. We encourage future works to explore the analysis of the robustness of unlearned models covering black-box jailbreak attacks.

3. Limiting update the model parameters w.r.t three layer $\{l, l-1, l-2\}$ thus risks missing interesting generalization behaviors.

7.2.2 Open Problems and Future Directions

Unlearning Evaluation. Measuring machine unlearning effectiveness is hard and still is an unresolved challenge. Current metrics often rely on downstream performance degradation on forget targets, but these are insufficient to capture whether knowledge has been fully removed or simply masked. Future work must explore more principled and comprehensive evaluation methods to establish reliable criteria for complete knowledge removal.

Robust Unlearning. MU methods must be robust not only to the distribution of inputs but also to adversarial attacks and relearning. Many current methods suffer from relearning and adversarial attacks. Building robust MU algorithms is an interesting and challenging task.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. (2024). Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493—-500.

Anthropic, A. (2024). The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1:1.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. (2022). Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Belrose, N., Schneider-Joseph, D., Ravfogel, S., Cotterell, R., Raff, E., and Biderman, S. (2023). Leace: Perfect linear concept erasure in closed form. *Advances in Neural Information Processing Systems*, 36:66044–66063.

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)*, pages 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*.

BUKATY, P. (2019). *The California Consumer Privacy Act (CCPA): An implementation guide*. IT Governance Publishing.

Cao, Y. and Yang, J. (2015). Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 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, pages 11186–11194.

Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J., and Wong, E. (2025). Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 23–42. IEEE.

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*, pages 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*.

Chris Jay Hoofnagle, B. v. d. S. and Borgesius, F. Z. (2019). The european union general data protection regulation: what it is and what it means*. *Information & Communications Technology Law*, 28(1):65–98.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

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*.

Dang, H.-T., Pham, T., Thanh-Tung, H., and Inoue, N. (2025). 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.

Deeb, A. and Roger, F. (2025). Do unlearning methods remove information from language model weights?

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*, pages 17108–17118.

Eldan, R. and Russinovich, M. (2023). Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*.

Eldan, R. and Russinovich, M. (2024). Who's harry potter? approximate unlearning for LLMs.

Fan, C., Liu, J., Lin, L., Jia, J., Zhang, R., Mei, S., and Liu, S. (2024a). Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *arXiv preprint arXiv:2410.07163*.

Fan, C., Liu, J., Zhang, Y., Wong, E., Wei, D., and Liu, S. (2024b). Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *The Twelfth International Conference on Learning Representations*.

Fang, R., Bindu, R., Gupta, A., Zhan, Q., and Kang, D. (2024). Llm agents can autonomously hack websites. *arXiv preprint arXiv:2402.06664*.

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, pages 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*, pages 2426–2436.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. (2024). The language model evaluation harness.

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*, pages 9304–9312.

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*.

He, L., Huang, Y., Shi, W., Xie, T., Liu, H., Wang, Y., Zettlemoyer, L., Zhang, C., Chen, D., and Henderson, P. (2024). Fantastic copyrighted beasts and how (not) to generate them. *arXiv preprint arXiv:2406.14526*.

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., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8759–8773. PMLR.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. (2020). Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

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*.

Hu, S., Fu, Y., Wu, S., and Smith, V. (2025). Unlearning or obfuscating? jogging the memory of unlearned LLMs via benign relearning. In *The Thirteenth International Conference on Learning Representations*.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., et al. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Huu-Tien, D., Thanh-Tung, H., Bui, A., Nguyen, L.-M., and Inoue, N. (2025). Improving llm unlearning robustness via random perturbations. *arXiv preprint arXiv:2501.19202*.

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., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 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*.

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*.

Karamolegkou, A., Li, J., Zhou, L., and Søgaard, A. (2023). Copyright violations and large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7403–7412.

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*, pages 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*.

Lee, A., Bai, X., Pres, I., Wattenberg, M., Kummerfeld, J. K., and Mihalcea, R. (2024). A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity. In *International Conference on Machine Learning*, pages 26361–26378. PMLR.

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., Mukobi, G., et al. (2024b). The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *International Conference on Machine Learning*, pages 28525–28550. PMLR.

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, pages 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*, pages 984–994.

Liu, J., Lou, J., Qin, Z., and Ren, K. (2024a). 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. (2024b). Towards safer large language models through machine unlearning. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1817–1829, Bangkok, Thailand. Association for Computational Linguistics.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Łucki, J., Wei, B., Huang, Y., Henderson, P., Tramèr, F., and Rando, J. (2024). An adversarial perspective on machine unlearning for ai safety. *arXiv preprint arXiv:2409.18025*.

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*.

Mantelero, A. (2013). The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law Security Review*, 29(3):229–235.

Merchant, A., Batzner, S., Schoenholz, S. S., Aykol, M., Cheon, G., and Cubuk, E. D. (2023). Scaling deep learning for materials discovery. *Nature*.

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., Ren, Z., 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.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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., Khanh, N. X., and Trinh, T. (2024). Probabilities of chat llms are miscalibrated but still predict correctness on multiple-choice q&a. *arXiv preprint arXiv:2402.13213*.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

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*.

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

Scholten, Y., Günnemann, S., and Schwinn, L. (2024). A probabilistic perspective on unlearning and alignment for large language models. *arXiv preprint arXiv:2410.03523*.

Shastri, S., Wasserman, M., and Chidambaram, V. (2019). The seven sins of personal-data processing systems under gdpr. In *Proceedings of the 11th USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'19, page 1, USA. USENIX Association.

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*.

Shi, W., Lee, J., Huang, Y., Malladi, S., Zhao, J., Holtzman, A., Liu, D., Zettlemoyer, L., Smith, N. A., and Zhang, C. (2025). MUSE: Machine unlearning six-way evaluation for language models. In *The Thirteenth International Conference on Learning Representations*.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. (2020). Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.

Sun, Y., Ming, Y., Zhu, X., and Li, Y. (2022). Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pages 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*, pages 489–492.

Thaker, P., Hu, S., Kale, N., Maurya, Y., Wu, Z. S., and Smith, V. (2025). Position: Llm unlearning benchmarks are weak measures of progress. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 520–533. IEEE.

Trinh, T., Wu, Y., Le, Q., He, H., and Luong, T. (2024). Solving olympiad geometry without human demonstrations. *Nature*.

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. (2025a). 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, page 622–632, New York, NY, USA. Association for Computing Machinery.

Wang, Y., Wei, J., Liu, C. Y., Pang, J., Liu, Q., Shah, A., Bao, Y., Liu, Y., and Wei, W. (2025b). LLM unlearning via loss adjustment with only forget data. In *The Thirteenth International Conference on Learning Representations*.

Wei, A., Haghtalab, N., and Steinhardt, J. (2023). Jailbroken: How does LLM safety training fail? In *Thirty-seventh Conference on Neural Information Processing Systems*.

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*, pages 23631–23644. PMLR.

Wei, R., Niu, P., Hsu, H. H.-H., Wu, R., Yin, H., Ghassemi, M., Li, Y., Potluru, V. K., Chien, E., Chaudhuri, K., et al. (2025). Do llms really forget? evaluating unlearning with knowledge correlation and confidence awareness. *arXiv preprint arXiv:2506.05735*.

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, page 2606–2617, New York, NY, USA. Association for Computing Machinery.

Wu, R., Yadav, C., Salakhutdinov, R., and Chaudhuri, K. (2024). Evaluating deep unlearning in large language models. *arXiv preprint arXiv:2410.15153*.

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., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 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).

Yamada, Y., Lange, R. T., Lu, C., Hu, S., Lu, C., Foerster, J., Clune, J., and Ha, D. (2025). The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*.

Yao, Y., Xu, X., and Liu, Y. (2023). Large language model unlearning. In *Socially Responsible Language Modelling Research*.

Yao, Y., Xu, X., and Liu, Y. (2024). Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475.

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, X., Pang, T., Du, C., Chen, K., Zhang, W., and Lin, M. (2025). A closer look at machine unlearning for large language models. In *The Thirteenth 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*, pages 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. (2023a). Recommendation unlearning via influence function. *ACM Transactions on Recommender Systems*.

Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., et al. (2023b). Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Zhang, Z., Wang, F., Li, X., Wu, Z., Tang, X., Liu, H., He, Q., Yin, W., and Wang, S. (2025). Catastrophic failure of LLM unlearning via quantization. In *The Thirteenth International Conference on Learning Representations*.

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

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

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., Carlini, N., Nasr, M., Kolter, J. Z., and Fredrikson, M. (2023b). Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.