Towards Robust Knowledge Unlearning: An Adversarial Framework for Assessing and Improving Unlearning Robustness in Large Language Models

Hongbang Yuan[1,2 *], **Zhuoran Jin**[1,2 *], **Pengfei Cao**[1,2], **Yubo Chen**[1,2 †], **Kang Liu**[1,2], **Jun Zhao**[1,2]

[1] The Key Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, China

[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

{hongbang.yuan, zhuoran.jin, pengfei.cao, yubo.chen , kliu, jzhao} @nlpr.ia.ac.cn

Abstract

LLM have achieved success in many fields but still troubled by problematic content in the training corpora. LLM unlearning aims at reducing their influence and avoid undesirable behaviours. However, existing unlearning methods remain vulnerable to adversarial queries and the unlearned knowledge resurfaces after the manually designed attack queries. As part of a red-team effort to proactively assess the vulnerabilities of unlearned models, we design **D**ynamic **U**nlearning **A**ttack (**DUA**), a dynamic and automated framework to attack these models and evaluate their robustness. It optimizes adversarial suffixes to reintroduce the unlearned knowledge in various scenarios. We find that unlearned knowledge can be recovered in 55.2% of the questions, even without revealing the unlearned model's parameters. In response to this vulnerability, we propose **L**atent **A**dversarial **U**nlearning (**LAU**), a universal framework that effectively enhances the robustness of the unlearned process. It formulates the unlearning process as a min-max optimization problem and resolves it through two stages: an attack stage, where perturbation vectors are trained and added to the latent space of LLMs to recover the unlearned knowledge, and a defense stage, where previously trained perturbation vectors are used to enhance unlearned model's robustness. With our LAU framework, we obtain two robust unlearning methods, **AdvGA** and **AdvNPO**. We conduct extensive experiments across multiple unlearning benchmarks and various models, and demonstrate that they improve the unlearning effectiveness by over 53.5%, cause only less than a 11.6% reduction in neighboring knowledge, and have almost no impact on the model's general capabilities.

1 Introduction

Large language models (LLMs) have achieved remarkable capabilities after being trained on an extensive amount of corpora (Chen et al. 2024; Sun et al. 2024). However, since the training data may contain copyrighted, private, and toxic content, LLMs inevitably learn some potentially undesirable behaviours (Ji et al. 2024). For example, LLMs may regurgitate copyrighted material without permission (Wei et al. 2024a), generate personal information such as phone numbers or mailing addresses (Yao et al. 2024b), and even produce offensive and harmful responses (Liu et al. 2024c). These unwanted behaviors introduce security concerns and information hazards, hindering the deployment of LLMs in real-world scenarios (Patil, Hase, and Bansal 2024).

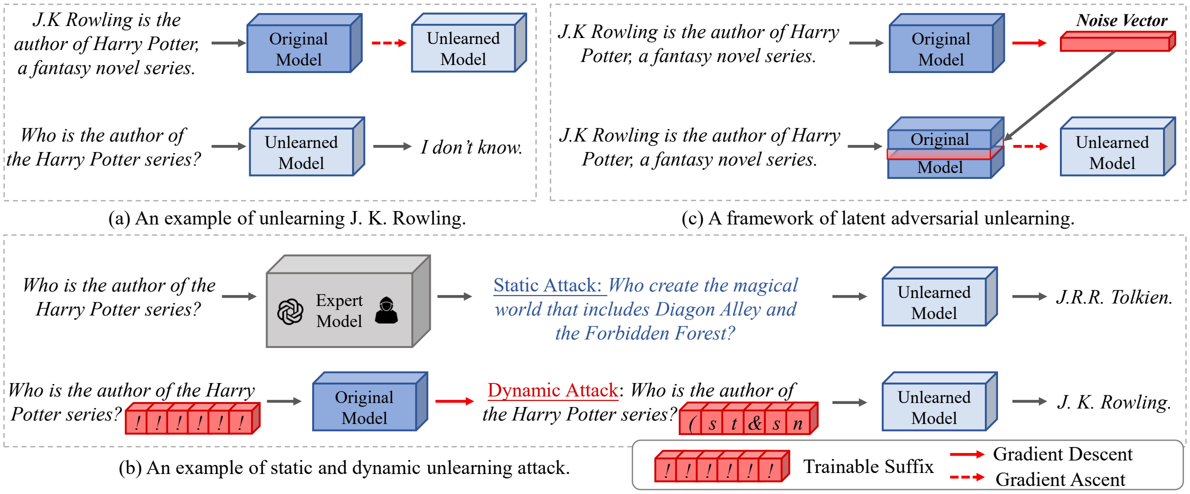
To eliminate the influence of problematic content in the corpora on LLMs, **machine unlearning** has emerged as a promising solution (Eldan and Russinovich 2023; Yao, Xu, and Liu 2024; Liu et al. 2024a). It transforms models to behave as if they were never trained on certain data entries so that specific target knowledge is erased, while other knowledge and capabilities of LLMs are preserved (Chen and Yang 2023; Maini et al. 2024). The most fundamental machine unlearning method is to adopt a **gradient ascent** procedure on the data that needs to be forgotten to fine-tune LLMs, reversing the original gradient descent optimization process (Jang et al. 2023; Yao et al. 2024a; Liu et al. 2024d).

However, despite their effectiveness, the unlearned models produced by these methods are fragile and susceptible to crafted adversarial user prompts (Patil, Hase, and Bansal 2024; Liu et al. 2024a). Particularly, the previously unlearned knowledge resurfaces through contextual interactions (Shumailov et al. 2024). For example, after being trained to forget the famous writer J.K. Rowling, the model fails to answer the question *'Who is the author of Harry Potter?'* but still outputs the correct answer to the question *'Who created the magical world that includes Diagon Alley and the Forbidden Forest?'*. To proactively assess the vulnerability of unlearned models to these malicious prompts, **static** and **manually-designed** adversarial prompts are used to reintroduce the unlearned knowledge (Jin et al. 2024). However, such a manual process is resource-intensive and often ineffective in guaranteeing the successful reintroduction of the previously unlearned knowledge.

Therefore, we propose a **dynamic** and **automated** attack framework, called **D**ynamic **U**nlearning **A**ttack (**DUA**), to quantitatively assess the robustness of the unlearned models. Specifically, as Figure 1(b) shows, we optimize a universal adversarial suffix that maximizes the probability of the model generating unlearned knowledge associated with a given unlearning target. This optimization process is performed and evaluated across various scenarios, considering both **practicality** and **generalization**. For practicality, we consider scenarios where the unlearned LLM is either accessible or inaccessible to the attacker. For generalization, we consider whether the adversarial suffix trained on certain

[*]These authors contributed equally to this work.

[†]Corresponding author



(a) An example of unlearning J. K. Rowling.

(c) A framework of latent adversarial unlearning.

(b) An example of static and dynamic unlearning attack.

Figure 1: Our work focuses on assessing the robustness of unlearned models by training adversarial suffixes and enhancing the robustness of the unlearning process through latent adversarial unlearning. In this figure, we show (a) An example of unlearning J.K.Rowling; (b) An example of static and dynamic unlearning attack; (c) A framework of latent adversarial unlearning.

questions about a specific unlearning target can be employed to other questions about the same target, or even to questions about different targets. For example, if a model is supposed to forget knowledge about J.K. Rowling, an adversarial suffix can be trained to make the model recall knowledge about her Harry Potter series. Then we test whether the same suffix remains effective when applied to questions about her book, The Cuckoo's Calling, or when in a model intended to forget knowledge about other authors. Experimental results demonstrate that the unlearned knowledge is recovered in 54% of the questions with the adversarial suffixes, even without disclosing the unlearned model to the attacker.

The revealed vulnerability motivates us to enhance the robustness of the unlearning process. Taking inspiration from previous work about adversarial training (Casper et al. 2024; Xhonneux et al. 2024), we propose a universal framework named **L**atent **A**dversarial **U**nlearning (**LAU**), which effectively enhances the robustness of the unlearned models and is inherently compatible with nearly all gradient-ascent-based unlearning methods. It consists of two optimizing processes: an attack process that aims to bring back the unlearned knowledge, and a defense process that strives to enhance the model's resistance to such attacks and suppress the recall of unlearned knowledge. For the attack process, as Figure 1(c) shows, we train perturbation vectors that will be added directly to in the **latent space** of LLMs to promote the unlearned knowledge. Particularly, we add a constrained perturbation vector to the hidden state at a specific layer of a LLM. This approach alleviates the burden of searching the vast input space and facilitates the rapid optimization of adversarial attacks. For the defense process, we fine-tune the model using the previously optimized perturbation vector in its latent space, thereby significantly improve the unlearned model's resistance to adversarial attacks.

To demonstrate the effectiveness of LAU, we propose **AdvGA** and **AdvNPO**, two robust variants of the widely used mainstream unlearning methods, GA and NPO. We conduct unlearning experiments with various models on two commonly used benchmarks, RWKU (Jin et al. 2024) and MUSE (Shi et al. 2024). Experimental results demonstrate that our LAU-augmented methods robustly forget the unlearning target with minimal side effects. Additionally, we assess the robustness of the LAU-trained models using our DUA framework and demonstrate that they exhibit greater resistance to adversarial attacks.

Our contributions can be summarized as follows:

- We propose **D**ynamic **U**nlearning **A**ttack (**DUA**), a dynamic and automated attack framework to quantitatively assess the robustness of the unlearned models. It recovers unlearned knowledge in 55.2% of the questions using trained adversarial suffixes, even without the direct access to the unlearned model itself.
- We propose **L**atent **A**dversarial **U**nlearning (**LAU**), a universal framework that effectively enhances the robustness of the unlearned process and is compatible with most gradient-ascent-based unlearning methods.
- We propose two robust unlearning methods, namely **AdvGA** and **AdvNPO**. Extensive experiments across multiple unlearning benchmarks and various models demonstrate that they improve the unlearning effectiveness by over 53.5%, cause only less than a 11.3% reduction in neighboring knowledge, and have almost no impact on the model's general capabilities.

2 Preliminaries

2.1 Problem Formulation

Given a LLM π_θ with parameter θ trained on dataset $D = \{(x_i, y_i) \mid i = 1, 2, \ldots, N\}$, we define the *forget set* D_f

as the specific training subset to be forgotten. Machine unlearning aims to eliminate the influence of D_f and make model π_θ behaves as if it is only trained on the *retain set* $D_r = D \setminus D_f$. Ideally, we can retrain the model on D_r from scratch but it is too costly and unrealistic thus effective approximate unlearning methods are essential. The most commonly used mathematical formulation for optimizing model unlearning is presented below:

$$\min_{\boldsymbol{\theta}} \underbrace{\mathbb{E}_{(x_{\mathrm{f}},y_{\mathrm{f}})\in\mathcal{D}_{\mathrm{f}}}\left[\ell_{\mathrm{f}}\left(y_{\mathrm{f}} \mid x_{\mathrm{f}};\boldsymbol{\theta}\right)\right]}_{\text{forget}} + \lambda \underbrace{\mathbb{E}_{(x_{\mathrm{r}},y_{\mathrm{r}})\in\mathcal{D}_{\mathrm{r}}}\ell_{\mathrm{r}}(y_{\mathrm{r}} \mid x_{\mathrm{r}};\boldsymbol{\theta})]}_{\text{retain}} \quad (1)$$

where ℓ_{f} and ℓ_{r} are the loss functions on forget set and retain set, respectively, and λ is a regularization parameter to balance them.

Typically, the forget set D_f and a subset of the retain set D_r are available in an unlearning task (Shi et al. 2024). In a more practical setting, only an unlearning target t is given and we shall generate a synthetic forget set D'_f related to this unlearning target t and a corresponding pseudo retain set D'_r (Jin et al. 2024). In our paper, both scenarios are considered.

2.2 Unlearning Methods

We introduce two widely used loss functions for the forget set, gradient ascent and negative preference optimization, and two widely used loss functions for the retain set, gradient descent and KL divergence.

Gradient Ascent. Gradient ascent servers as an important baseline method in machine unlearning and uses the following loss function:

$$\ell_{\mathrm{f}} = -\mathbb{E}_{D_{\mathrm{f}}} L\left(\pi_\theta(x,y)\right) \quad (2)$$

where L represents a cross-entropy prediction loss. It aims to maximize the average prediction loss of LLMs on the forget dataset, thereby reverting the original gradient descent training process on the forget set.

Negative Preference Optimization. Due to the divergent nature of the gradient ascent method, the unlearning process suffers from catastrophic collapse, wherein the model quickly deteriorates and generates incoherent responses (Zhang et al. 2024). Thus, the negative preference optimization algorithm is introduced, characterized by the following loss function:

$$\ell_{\mathrm{f}} = -\frac{2}{\beta}\mathbb{E}_{D_{\mathrm{f}}}\left[\log\left(1+\left(\frac{L(\pi_\theta(x),y)}{L\left(\pi_{ref}(x,y)\right)}\right)^{\beta}\right)\right] \quad (3)$$

where β is a regularization parameter controlling the deviation between the current model π_θ and the original model π_{ref}. It provides more stable training dynamics and achieves better performance.

Gradient Descent. One widely adopted loss function for the retain set is the standard cross-entropy loss function:

$$\ell_{\mathrm{r}} = \mathbb{E}_{D_{\mathrm{r}}} L\left(\pi_\theta(x,y)\right) \quad (4)$$

KL Divergence. Another approach is to minimize the Kullback-Leibler (KL) divergence between the predictions of the original model and the current model on the retain dataset. The loss function can be expressed as:

$$\ell_{\mathrm{r}} = \mathbb{E}_{D_{\mathrm{r}}} D_{KL}(\pi_\theta(x,y)||\pi_{ref}(x,y)) \quad (5)$$

where π_θ and π_{ref} denote the current model and the original reference model, respectively. It prevents the model from collapsing by ensuring that it does not deviate excessively from the original predictions.

2.3 Datasets and Metrics

RWKU. RWKU (Jin et al. 2024) is a real-world knowledge unlearning benchmark that requires models to erase specific knowledge in their parameters. Specifically, for the evaluation of *unlearning effectiveness*, it provides three types of knowledge probe questions on the forget set: FB, QA and AA. For the evaluation of *utility preservation*, it provides two types of questions on the neighbor set to test the impact of neighbor perturbation: FB and QA. We use ROUGE-L score (Lin 2004) to measure model's performance. Lower scores on forget set indicate better unlearning effectiveness, and higher scores on neighbor set indicate better utility preservation. Additionally, it measures the model's various capabilities, including reasoning (Rea), truthfulness (Tru), factuality (Fac) and fluency (Flu). It also provides some membership inference attacks (MIA) to detect retained knowledge in LLMs, where higher scores indicate that the model retains specific knowledge. Further details and examples are presented in Appendix A.

MUSE. MUSE (Shi et al. 2024) is a comprehensive unlearning benchmark that requires models to unlearn either news articles or book series. Similarly, it also contains the evaluation for unlearning effectiveness and utility preservation. More details are presented in Appendix A.

3 Dynamic Unlearning Attack Framework

In this section, we introduce a dynamic, automated framework to assess the robustness of the unlearned models. Firstly, we describe the process for optimizing adversarial suffixes that reintroduce the unlearned knowledge. Subsequently, we introduce the various attack scenarios, focusing on both practicality and generalization. Finally, we conduct experiments on ten unlearned models, demonstrating that they remain susceptible to adversarial attacks even without exposing their parameters.

3.1 Adversarial Suffix Optimization

Motivated by the GCG attack in safety-related domains (Zou et al. 2023), we introduce how to optimize attack suffixes in the context of unlearning. Intuitively, we optimize the suffix tokens to maximize the probability of generating the unlearned knowledge.

Consider a question $x_{[0,m)]}$ related to the unlearning target. We aim to find an adversarial suffix $q_{[0,n)}$ that, when combined with x to form $[x; q]$, makes the unlearning model

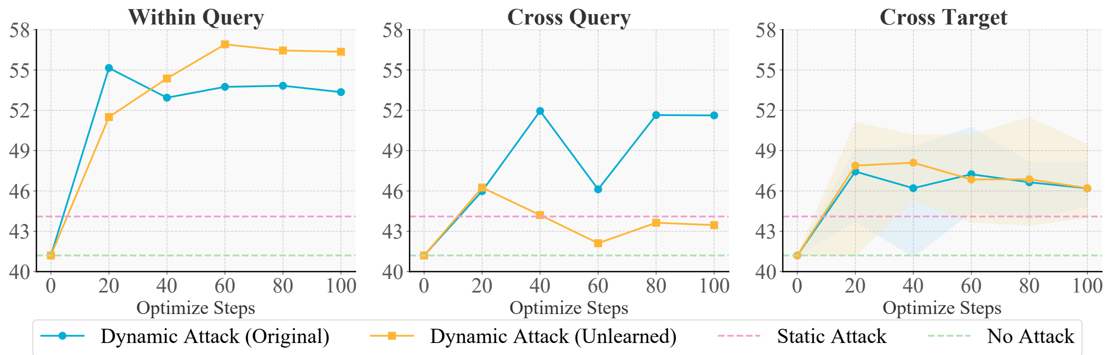


Figure 2: Experimental results of our dynamic attack framework. We report the ROUGE-L recall score (%) in this figure.

generate a response $y_{[0,H)}$ containing the unlearned knowledge. The optimization process can be expressed as:

$$\min_{q_{[0,n)} \in \{1,..,V\}} -log \prod_{i=0}^{H-1} p\left(y_{i+1} \mid [x_{0:m}; q_{0:n}; y_{0:i}]\right) \tag{6}$$

where p denotes the next token prediction probability, V denotes the vocabulary size, m is the number of tokens in user query x, n is the number of tokens in the adversarial suffix q, and y is the desired response.

To solve this optimization problem, at each optimization step, we leverage the gradients of the tokens to identify a set of candidate tokens, then evaluate them token by token and select the optimal one. Practically, we optimize one single suffix across multiple prompts, resulting in a universal suffix that can be transferred to other queries.

3.2 Robustness Assessment

The optimization process defined by Equation 6 can be performed and evaluated in various scenarios. Depending on the choice of the training data (x, y) and the computation of the next token prediction probability p, we design our assessment framework from the perspective of practicality and generalization.

Practicality. We consider two scenarios for the calculation of the next token probability p. (1) *Attack Unlearned.* The ideal approach is to use the unlearned model, as the adversarial suffix will ultimately be used to attack this model. (2) *Attack Original.* We also consider a more practical scenario when the unlearned model is not available to the attacker. Therefore, we directly use the models before unlearning to optimize the adversarial suffixes.

Generalization. Typically, the ability of the unlearned models can be assessed by the question-answer style probes related to the unlearning target. Thus the training data (x, y) should be specified as similar question-answer pairs. This raises the question of whether the testing questions are available to the attacker, and whether the unlearning target itself is accessible. We consider three settings, each of which imposes progressively higher demands for generalization.

(1) *Within Query.* The test questions are available to the attacker, thereby we can directly train adversarial queries on the testing questions. (2) *Cross Query.* The test questions are not available to the attacker, necessitating the generation of training data based on the unlearning target. (3) *Cross Target.* The unlearning target itself is not available to the attacker and the training data must be obtained using knowledge probe questions about other unlearning targets.

3.3 Experiments

Configuration. To assess the robustness of the unlearned models with our framework, we first conduct unlearning experiments with Llama-3-8B-Instruct on dataset RWKU using the negative preference optimization method. Subsequently, we apply our attack framework to create adversarial queries across various scenarios and evaluate the performance of the unlearned models. We present the average performance of 10 models, each trained with unlearn different targets. For the *cross query* setting, we generate knowledge probe questions for training based on the unlearning target using GPT-4. For the *cross target* setting, we use an additional 5 unlearned models along with their corresponding knowledge probe questions to train the adversarial queries.

Additionally, we generate an equivalent number of static attack questions using GPT-4 for comparison. Details of the construction process for the static attack questions, along with specific examples of both dynamic and static attack questions, are provided in Appendix B.

Results. The experimental results are presented in Figure 2. We can draw the following conclusions: (1) The unlearned models are vulnerable to adversarial queries, especially when the unlearned models and test queries are accessible to the attacker. For example, the unlearned model demonstrates a maximum performance increase of 15.25% using adversarial queries compared to not using them. It indicates that the previously forgotten knowledge gets reintroduced in-context. (2) Models are more resistant to attacks when knowledge probe queries and unlearning targets are inaccessible to attackers. However, our dynamic framework is still able to improve model performance beyond that of static attacks, highlighting the limitations of the resistance. (3) Even without access to the unlearned models, the attacker can still carry out attacks that are nearly as effective as those conducted with access to the unlearned models. For example, the '*attack original*' lines are almost equivalent to the '*attack unlearned*' lines and even outperform them in the *cross query* setting. This reveals a more critical issue that has been overlooked: malicious queries can be trained to recover forgotten knowledge even without prior access to the unlearned model itself.

4 Latent Adversarial Unlearning Framework

In this section, we propose an adversarial learning framework to increase the robustness of the unlearning process. First, we will propose a saddle-point (min-max) formulation for adversarial training in the context of machine unlearning, and subsequently, we will elaborate how our framework can be employed to concrete methods in detail.

4.1 Framework Formulation

In the context of unlearning, we formulate the adversarial training as the following min-max optimization problem:

$$\max_{\theta} \mathbb{E}_{(x_{\mathrm{f}},y_{\mathrm{f}})\in\mathcal{D}_{\mathrm{f}}}\left[\min_{\epsilon\in T(x_{\mathrm{f}})} \mathcal{L}\left(\pi_\theta(x_{\mathrm{f}}+\epsilon),y\right)\right] \tag{7}$$

where $\mathcal{L}$ is a negative cross-entropy loss function and $T(x_{\mathrm{f}})$ is a set of adversarial perturbations generated by various attack methods. It's a composition of an *inner minimization* problem and an *outer maximization* problem. The inner minimization process aims to identify adversarial queries that effectively bypass the model's restrictions and activate the forgotten knowledge. The outer maximization strives to suppress the re-emergence of the forgotten knowledge on these adversarial queries.

However, it is non-trivial to identify all the elements in the adversarial query set $T(x)$, as there are too many potential adversarial queries hidden beneath. Additionally, optimizing a discrete set of tokens is challenging. To avoid the intricate process of optimizing adversarial queries, we propose *latent adversarial unlearning*. The core idea is that any type of adversarial query will cause a perturbation in the latent space of LLMs, potentially leading to the resurgence of forgotten knowledge. Therefore, we directly add perturbations to the latent space of LLMs, thus avoiding the extensive optimization in the input space. This process can be formulated as the following min-max optimization problem:

$$\max_{\theta} \mathbb{E}_{\mathcal{D}_{\mathrm{f}}}\left[\min_{\delta} \mathcal{L}\left(\pi_{\theta_2}(\pi_{\theta_1}(x_{\mathrm{f}})+\delta_{x_{\mathrm{f}}}),y_{\mathrm{f}}\right)\right] s.t.\|\delta_{x_{\mathrm{f}}}\| \le \kappa \tag{8}$$

where π_{θ_1} and π_{θ_2} represent the computations in LLM π_θ before and after the perturbation δ is added, respectively. The L_2-norm of the perturbation vector is restricted to a constant κ. In this way, both the inner and outer optimization problems can be solved using gradient descent algorithms.

Practically, we add a perturbation vector to the residual stream at a specific layer of a transformer model. For each batch of samples, we optimize the perturbation vector using its gradient for a fixed number of steps. Subsequently, we apply the classical stochastic gradient descent algorithm to update the model's parameters, with the previously optimized perturbation vector in its residual streams. The impact of the choice of perturbation layers and the number of inner optimization steps will be discussed in Section 5.

4.2 Two Adversarial Unlearning Methods

Our adversarial unlearning framework is suitable for most of the existing machine unlearning algorithms. In this paper, we apply our framework to augment the GA and NPO methods, resulting in two new algorithms: AdvGA and AdvNPO, which are described below.

AdvGA. By substituting the internal minimization loss function in Equation 8 with Equation 3, we can obtain the following loss function [1]:

$$\min_{\pi_\theta} -\mathbb{E}_{D_{\mathrm{f}}} \min_{\delta} L\left(\pi_{\theta_2}(\pi_{\theta_1}(x)+\delta,y)\right) \tag{9}$$

We denote this new loss function *AdvGA*.

AdvNPO. Similarly, we substitute the internal minimization loss function in equation 8 with Equation 3 and the following loss function is obtained:

$$\min_{\pi_\theta} -\frac{2}{\beta}\mathbb{E}_{D_{\mathrm{f}}}\left[\log\left(1+\left(\frac{\min_\delta L\left(\pi_{\theta_2}(\pi_{\theta_1}(x)+\delta,y)\right)}{L\left(\pi_{ref}(x,y)\right)}\right)^{\beta}\right)\right] \tag{10}$$

We denote this new loss function $AdvNPO$. For clarification, we omit the L_2-norm restriction on the perturbation vector here, but it should be included during the optimization process. This also applies to the loss function in AdvGA.

4.3 Experiments

Configurations. We conduct machine unlearning experiments on the following two datasets: RWKU (Jin et al. 2024) and MUSE (Shi et al. 2024). We combine the previously introduced forget loss functions and retain loss functions, and finally obtain 12 unlearning methods as shown in Table 1. We set the perturbation layer to 4, the inner optimization steps to 6, and the weights of the forget and retain loss functions to be equal. We conduct experiments with LLaMA-2-7B-Chat (Touvron et al. 2023), LLaMA-3-8B-Instruct and LLaMA-3.1-8B-Instruct. Following previous work, we run the optimizing process using the AdamW optimizer with a cosine learning rate scheduler. All the experiments are conducted on 4 Nvidia A100 GPUs. Further details are provided in Appendix C.

Results. The experimental results on RWKU with LLaMA-3-8B-Instruct and LLaMA-3.1-8B-Instruct are presented in Table 1. Additional results with other models on MUSE are provided in the Appendix C. From the table, we can draw the following conclusions:

(1) Our methods, particularly the AdvNPO series, are highly effective in unlearning the real-world knowledge in LLMs. For example, $\text{AdvNPO}_{\text{KLR}}$ achieves a performance increase of 57% on the forget set, but only causes a drop of 10.6% on the neighbor set comparing with a vanilla NPO method. This demonstrates the effectiveness of the LAU

[1] It should be noted that the outer maximization is converted to minimization by introducing a negative sign, thereby maintaining consistency with the loss minimization format.

Methods	Forget Set ↓				Neighbor Set ↑			MIA Set		Utility Set ↑			
	FB	QA	AA	All	FB	QA	All	FM ↑	RM ↓	Rea	Tru	Fac	Flu
LLaMA-3-8B-Instruct													
Before	85.6	70.3	74.7	76.9	93.1	82.0	87.6	236.5	230.9	41.0	36.4	53.7	704.6
GA	72.0	64.6	68.5	68.4	85.0	74.7	79.8	241.4	234.6	40.4	37.6	49.6	710.3
AdvGA	63.0	48.2	60.5	57.2 $^{\downarrow 16.4\%}$	75.8	72.1	74.0 $^{\downarrow 7.3\%}$	202.0	176.5	40.1	35.2	49.4	717.0
GA_{GDR}	72.6	64.0	69.7	68.8	86.2	76.5	81.4	242.8	236.8	39.6	36.8	50.4	710.3
$AdvGA_{GDR}$	69.2	52.4	66.1	62.6 $^{\downarrow 9.0\%}$	85.7	73.7	79.7 $^{\downarrow 2.1\%}$	205.2	184.5	41.4	35.4	50.5	712.1
GA_{KLR}	70.7	57.5	69.9	66.1	80.5	70.5	75.5	242.4	230.8	41.5	35.6	54.0	704.4
$AdvGA_{KLR}$	58.8	43.8	59.5	54.0 $^{\downarrow 18.3\%}$	76.9	63.0	69.9 $^{\downarrow 7.4\%}$	371.3	340.8	41.2	33.8	50.5	712.6
NPO	46.6	39.0	35.3	40.3	79.2	70.9	75.1	263.3	241.4	40.5	36.0	56.7	695.9
AdvNPO	19.7	14.7	12.0	15.5 $^{\downarrow 61.5\%}$	67.0	59.7	63.3 $^{\downarrow 15.7\%}$	270.1	238.9	39.3	34.0	56.8	663.1
NPO_{GDR}	52.2	43.9	42.9	46.3	82.5	70.5	76.5	254.5	240.1	39.6	37.2	51.4	708.2
$AdvNPO_{GDR}$	25.5	22.1	16.5	21.4 $^{\downarrow 53.8\%}$	71.9	69.1	70.5 $^{\downarrow 7.8\%}$	248.8	223.1	41.9	35.8	52.4	705.2
NPO_{KLR}	52.5	40.6	43.2	45.4	83.2	72.1	77.6	253.0	236.9	40.9	35.4	54.2	704.9
$AdvNPO_{KLR}$	23.6	18.9	16.0	19.5 $^{\downarrow 57.0\%}$	72.1	66.8	69.4 $^{\downarrow 10.6\%}$	347.2	318.1	41.7	35.6	55.3	697.1
LLaMA-3.1-8B-Instruct													
Before	63.9	65.1	69.5	66.2	74.1	69.8	72.0	223.5	218.2	42.2	35.4	61.2	695.2
GA	50.7	45.4	61.2	52.4	45.6	37.2	41.4	248.9	241.9	43.2	35.8	48.7	726.6
AdvGA	32.0	22.5	36.0	30.2 $^{\downarrow 42.4\%}$	27.5	21.0	24.3 $^{\downarrow 41.3\%}$	173.7	125.9	39.8	33.0	28.8	730.1
GA_{GDR}	55.4	49.6	63.9	56.3	60.2	53.5	56.9	239.8	231.3	44.2	35.0	53.9	718.5
$AdvGA_{GDR}$	44.0	34.1	47.8	42.0 $^{\downarrow 25.4\%}$	62.6	52.5	57.6 $^{\uparrow 1.2\%}$	71.9	62.3	43.2	35.8	52.7	718.6
GA_{KLR}	62.7	49.9	66.4	59.7	67.9	61.2	64.6	235.8	223.0	42.6	35.4	59.0	682.1
$AdvGA_{KLR}$	50.8	42.0	54.8	49.2 $^{\downarrow 17.6\%}$	59.8	59.8	59.8 $^{\downarrow 7.4\%}$	69.1	67.2	43.1	33.4	57.3	697.5
NPO	35.7	40.2	39.0	38.3	67.3	66.2	66.7	241.4	220.5	42.5	35.6	61.8	684.2
AdvNPO	18.0	21.7	16.5	18.7 $^{\downarrow 51.2\%}$	60.0	57.2	58.6 $^{\downarrow 12.1\%}$	108.3	86.9	41.1	35.4	61.4	677.8
NPO_{GDR}	42.4	37.2	42.0	40.5	74.0	66.7	70.3	236.3	220.1	43.0	35.4	60.8	698.8
$AdvNPO_{GDR}$	23.1	20.8	16.7	20.2 $^{\downarrow 50.1\%}$	62.4	59.7	61.1 $^{\downarrow 13.1\%}$	91.0	77.6	42.6	35.4	60.7	696.1
NPO_{KLR}	40.6	41.4	42.2	41.4	73.3	69.9	71.6	234.4	218.8	42.3	35.4	61.5	695.1
$AdvNPO_{KLR}$	24.1	18.5	19.4	20.7 $^{\downarrow 50.0\%}$	65.0	61.0	63.0 $^{\downarrow 12.0\%}$	88.9	74.9	42.2	35.2	60.5	690.2

Table 1: Experimental results on RWKU with LLaMA-3-8B-Instruct and LLaMA-3.1-8B-Instruct. The **superscript** denotes the performance increase of our adversarial methods compared to the corresponding non-adversarial versions. Please refer to Section 2.3 for the meaning of the **abbreviations**.

framework. (2) Our methods cause almost no side effects on the general capabilities of LLMs. For example, performance on the utility set remains nearly the same before and after the unlearning process. This demonstrates that the unlearned models remain powerful in many scenarios, despite having specific knowledge removed.

5 Discussion

5.1 Influence of the Perturb Layers

We explore how the specific layer at which the perturbation vector is added influences the final performance. Therefore, we vary the perturbation layer from 0 to 30 for the Llama-3-8B-Instruct model and report the averaged performance on the forget and neighbor datasets for one unlearned model. The experimental results are shown in the upper half of Figure 3. We can draw the following conclusions.

(1) Adding perturbations at the shallow layers (those closer to the input prompts) is more effective. We attribute this to the fact that perturbations at shallower layers have a more significant impact on the model's output, making them easier to optimize. (2) Directly adding perturbation at the embedding layer is entirely ineffective, as indicated by the point where the perturbation layer equals zero. This is due to the fact that our latent perturbation serves as an approximation of the adversarial queries, but directly adding perturbation at the embedding layer alters the entire prompt rather than simply adding an adversarial suffix. (3) As the perturbation layers get deeper, the performance converges to that of the corresponding non-adversarial method. This finding aligns with the intuition that deeper perturbation layers result in a more limited influence of the perturbation vectors.

5.2 Influence of the Inner Optimization Steps

Similarly, we also explore the influence of the number of inner optimization steps in Equation 8. We follow the same experimental configurations as before, but instead vary the number of inner optimization steps. The experimental results are shown in the lower half of Figure 3. We can draw the following conclusions:

(1) As we increase the optimization steps, the performance on the forget dataset initially declines, then rises. We thereby conclude that both insufficient and excessive optimization steps are detrimental to unlearning performance. (2) Regardless of the number of optimization steps, the model's performance on the forget dataset consistently remains below that of non-adversarial methods. Even a small, randomly initialized perturbation vector can enhance the robustness of the unlearning process, as indicated by the point where the step equals 0.

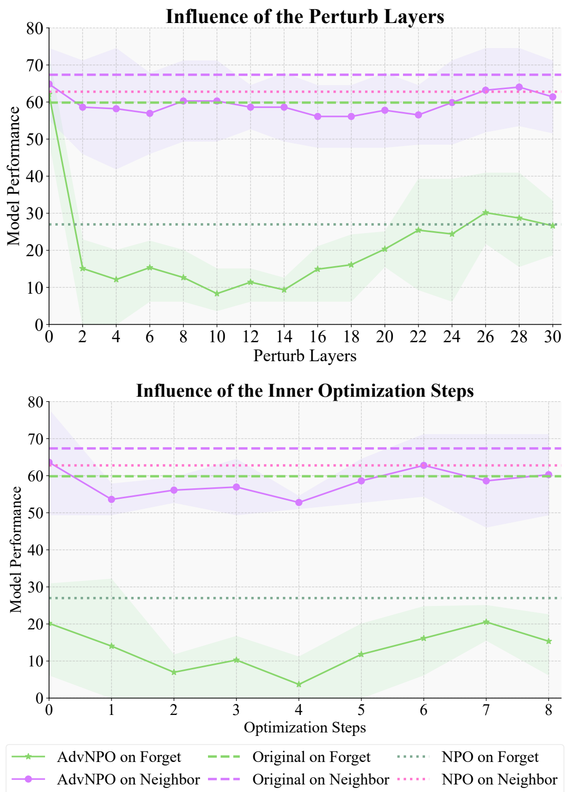


Figure 3: Influence of the perturb layers and the inner optimization steps. We report the ROUGE-L recall score (%).

5.3 Robustness of Latent Adversarial Unlearning

Finally, we evaluate the robustness of the unlearned models trained with LAU-augmented methods under our previously proposed dynamic attack framework. We select ten unlearned models and train adversarial suffixes with both the original model and the unlearned model in the *within query* setting. For a clearer comparison with unlearned models trained using non-adversarial approaches, we report the performance change relative to the scenario without attack. The experimental results are presented in Figure 4. We can draw the following conclusions:

(1) The LAU-trained models are more robustness than the non-LAU-trained models. For instance, the increased performance under adversaries of the model trained with NonLAU is nearly twice that of the model trained with LAU. This demonstrates a significant enhancement in the robustness of the unlearned models. (2) The unlearned models become safer especially when their parameters are not available to the attacker. For example, the 'Attack LAU (original)' line consistently remains at a low value as the attack optimization steps increase.

6 Related Work

Jailbreak Attack. To mitigate the undesirable behaviors of LLMs, the safety-alignment stage has become essential (Wei, Haghtalab, and Steinhardt 2023; Paulus et al. 2024;

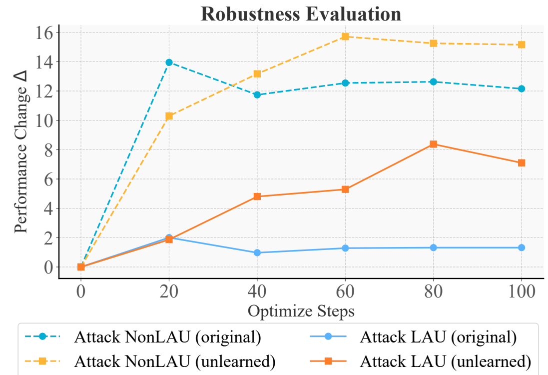


Figure 4: Robustness evaluation of AdvNPO. We report the performance change (Δ) in terms of the ROUGE-L recall score (%) compared to the scenario without attack.

Tamirisa et al. 2024). Within this context, a complementary approach called *red teaming* is proposed to assess the robustness of the safety-alignment of LLMs by designing *jailbreaking* attacks (Carlini et al. 2023; Huang et al. 2024). Some work focuses on designing manually crafted adversarial prompts (Yong, Menghini, and Bach 2024; Wei et al. 2024b), while others explore automatically generating the prompts via gradient-based optimization (Jones et al. 2023; Zou et al. 2023) or genetic-base methods (Lapid, Langberg, and Sipper 2023; Liu et al. 2024b). However, they primarily focus on the introduction of harmful behaviors, whereas we focus on the resurgence of unlearned knowledge.

Machine Unlearning. Data protection regulations, such as the European General Data Protection Regulation (GDPR) (Mantelero 2013), have mandated "the Right to be Forgotten" and highlight the necessity of *machine unlearning* (Hu et al. 2024; Schwinn et al. 2024; Sheshadri et al. 2024). Therefore, a number of unlearning benchmarks are proposed, including the forgetting of Harry Potter books (Eldan and Russinovich 2023), facts about fictional authors (Maini et al. 2024) and real world knowledge in LLMs (Jin et al. 2024). Additionally, a line of research explores the methodologies for effective machine unlearning, from variants of gradient-ascent (Jia et al. 2023; Zhang et al. 2024), to localization-informed approaches (Wu et al. 2023; Fan et al. 2024; Rosati et al. 2024). However, existing unlearning methods remain vulnerable and our proposed LAU framework provides a universal solution to enhancing the robustness of the unlearning process.

7 Conclusion

In this paper, we propose a dynamic and automated framework to assess the vulnerabilities of the unlearned models. After revealing their susceptibility, we propose a latent adversarial training framework, along with two concrete methods, namely AdvGA and AdvNPO. Extensive experiments on several datasets with various models demonstrate the effectiveness and robustness of our unlearning methods.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 62176257, No.U24A20335, No.62406321). This work is also supported by Beijing Natural Science Foundation (L243006). This work is also supported by the Youth Innovation Promotion Association CAS and the China Postdoctoral Science Foundation under Grant Number 2024M753500.

References

Carlini, N.; Nasr, M.; Choquette-Choo, C. A.; Jagielski, M.; Gao, I.; Koh, P. W.; Ippolito, D.; Tramèr, F.; and Schmidt, L. 2023. Are aligned neural networks adversarially aligned? In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Casper, S.; Schulze, L.; Patel, O.; and Hadfield-Menell, D. 2024. Defending Against Unforeseen Failure Modes with Latent Adversarial Training. arXiv:2403.05030.

Chen, H.; Jiao, F.; Li, X.; Qin, C.; Ravaut, M.; Zhao, R.; Xiong, C.; and Joty, S. 2024. ChatGPT's One-year Anniversary: Are Open-Source Large Language Models Catching up? arXiv:2311.16989.

Chen, J.; and Yang, D. 2023. Unlearn What You Want to Forget: Efficient Unlearning for LLMs. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 12041–12052. Singapore: Association for Computational Linguistics.

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

Fan, C.; Liu, J.; Zhang, Y.; Wong, E.; Wei, D.; and Liu, S. 2024. SalUn: Empowering Machine Unlearning via Gradient-based Weight Saliency in Both Image Classification and Generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Hu, S.; Fu, Y.; Wu, Z. S.; and Smith, V. 2024. Jogging the Memory of Unlearned Model Through Targeted Relearning Attack. arXiv:2406.13356.

Huang, Y.; Gupta, S.; Xia, M.; Li, K.; and Chen, D. 2024. Catastrophic Jailbreak of Open-source LLMs via Exploiting Generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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

Ji, J.; Qiu, T.; Chen, B.; Zhang, B.; Lou, H.; Wang, K.; Duan, Y.; He, Z.; Zhou, J.; Zhang, Z.; Zeng, F.; Ng, K. Y.; Dai, J.; Pan, X.; O'Gara, A.; Lei, Y.; Xu, H.; Tse, B.; Fu, J.; McAleer, S.; Yang, Y.; Wang, Y.; Zhu, S.-C.; Guo, Y.; and Gao, W. 2024. AI Alignment: A Comprehensive Survey. arXiv:2310.19852.

Jia, J.; Liu, J.; Ram, P.; Yao, Y.; Liu, G.; Liu, Y.; Sharma, P.; and Liu, S. 2023. Model Sparsity Can Simplify Machine Unlearning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Jin, Z.; Cao, P.; Wang, C.; He, Z.; Yuan, H.; Li, J.; Chen, Y.; Liu, K.; and Zhao, J. 2024. RWKU: Benchmarking Real-World Knowledge Unlearning for Large Language Models. arXiv:2406.10890.

Jones, E.; Dragan, A. D.; Raghunathan, A.; and Steinhardt, J. 2023. Automatically Auditing Large Language Models via Discrete Optimization. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, 15307–15329. PMLR.

Lapid, R.; Langberg, R.; and Sipper, M. 2023. Open Sesame! Universal Black Box Jailbreaking of Large Language Models. arXiv:2309.01446.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Liu, S.; Yao, Y.; Jia, J.; Casper, S.; Baracaldo, N.; Hase, P.; Yao, Y.; Liu, C. Y.; Xu, X.; Li, H.; Varshney, K. R.; Bansal, M.; Koyejo, S.; and Liu, Y. 2024a. Rethinking Machine Unlearning for Large Language Models. arXiv:2402.08787.

Liu, X.; Xu, N.; Chen, M.; and Xiao, C. 2024b. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models. arXiv:2310.04451.

Liu, Y.; Deng, G.; Xu, Z.; Li, Y.; Zheng, Y.; Zhang, Y.; Zhao, L.; Zhang, T.; Wang, K.; and Liu, Y. 2024c. Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study. arXiv:2305.13860.

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

Maini, P.; Feng, Z.; Schwarzschild, A.; Lipton, Z. C.; and Kolter, J. Z. 2024. TOFU: A Task of Fictitious Unlearning for LLMs. arXiv:2401.06121.

Mantelero, A. 2013. Corrigendum to "The EU Proposal for a General Data Protection Regulation and the roots of the 'right to be forgotten'" [2013] 29 CLSR 229-235. *Comput. Law Secur. Rev.*, 29(5): 637.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Paulus, A.; Zharmagambetov, A.; Guo, C.; Amos, B.; and Tian, Y. 2024. AdvPrompter: Fast Adaptive Adversarial Prompting for LLMs. arXiv:2404.16873.

Rosati, D.; Wehner, J.; Williams, K.; Łukasz Bartoszcze; Atanasov, D.; Gonzales, R.; Majumdar, S.; Maple, C.; Sajjad, H.; and Rudzicz, F. 2024. Representation Noising: A Defence Mechanism Against Harmful Finetuning. arXiv:2405.14577.

Schwinn, L.; Dobre, D.; Xhonneux, S.; Gidel, G.; and Gunnemann, S. 2024. Soft Prompt Threats: Attacking Safety Alignment and Unlearning in Open-Source LLMs through the Embedding Space. arXiv:2402.09063.

Sheshadri, A.; Ewart, A.; Guo, P.; Lynch, A.; Wu, C.; Hebbar, V.; Sleight, H.; Stickland, A. C.; Perez, E.; Hadfield-Menell, D.; and Casper, S. 2024. Latent Adversarial Training Improves Robustness to Persistent Harmful Behaviors in LLMs. arXiv:2407.15549.

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

Shumailov, I.; Hayes, J.; Triantafillou, E.; Ortiz-Jimenez, G.; Papernot, N.; Jagielski, M.; Yona, I.; Howard, H.; and Bagdasaryan, E. 2024. UnUnlearning: Unlearning is not sufficient for content regulation in advanced generative AI. arXiv:2407.00106.

Sun, T.; Zhang, X.; He, Z.; Li, P.; Cheng, Q.; Liu, X.; Yan, H.; Shao, Y.; Tang, Q.; Zhang, S.; Zhao, X.; Chen, K.; Zheng, Y.; Zhou, Z.; Li, R.; Zhan, J.; Zhou, Y.; Li, L.; Yang, X.; Wu, L.; Yin, Z.; Huang, X.; Jiang, Y.-G.; and Qiu, X. 2024. MOSS: An Open Conversational Large Language Model. *Machine Intelligence Research*, 21(5): 888–905.

Tamirisa, R.; Bharathi, B.; Phan, L.; Zhou, A.; Gatti, A.; Suresh, T.; Lin, M.; Wang, J.; Wang, R.; Arel, R.; Zou, A.; Song, D.; Li, B.; Hendrycks, D.; and Mazeika, M. 2024. Tamper-Resistant Safeguards for Open-Weight LLMs. arXiv:2408.00761.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Wei, A.; Haghtalab, N.; and Steinhardt, J. 2023. Jailbroken: How Does LLM Safety Training Fail? In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Wei, B.; Shi, W.; Huang, Y.; Smith, N. A.; Zhang, C.; Zettlemoyer, L.; Li, K.; and Henderson, P. 2024a. Evaluating Copyright Takedown Methods for Language Models. arXiv:2406.18664.

Wei, Z.; Wang, Y.; Li, A.; Mo, Y.; and Wang, Y. 2024b. Jailbreak and Guard Aligned Language Models with Only Few In-Context Demonstrations. arXiv:2310.06387.

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

Xhonneux, S.; Sordoni, A.; Günnemann, S.; Gidel, G.; and Schwinn, L. 2024. Efficient Adversarial Training in LLMs with Continuous Attacks. arXiv:2405.15589.

Yao, J.; Chien, E.; Du, M.; Niu, X.; Wang, T.; Cheng, Z.; and Yue, X. 2024a. Machine Unlearning of Pre-trained Large Language Models. arXiv:2402.15159.

Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; and Zhang, Y. 2024b. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 100211.

Yao, Y.; Xu, X.; and Liu, Y. 2024. Large Language Model Unlearning. arXiv:2310.10683.

Yong, Z.-X.; Menghini, C.; and Bach, S. H. 2024. Low-Resource Languages Jailbreak GPT-4. arXiv:2310.02446.

Zhang, R.; Lin, L.; Bai, Y.; and Mei, S. 2024. Negative Preference Optimization: From Catastrophic Collapse to Effective Unlearning. arXiv:2404.05868.

Zou, A.; Wang, Z.; Carlini, N.; Nasr, M.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. arXiv:2307.15043.