Refusal Is Not an Option: Unlearning Safety Alignment of Large Language Models

Minkyoo Song Hanna Kim Jaehan Kim Seungwon Shin Sooel Son
KAIST, South Korea
{minkyoo9, gkssk3654, jaehan, claude, sl.son}@kaist.ac.kr

Abstract

Safety alignment has become an indispensable procedure to ensure the safety of large language models (LLMs), as they are reported to generate harmful, privacy-sensitive, and copyrighted content when prompted with adversarial instructions. Machine unlearning is a representative approach to establishing the safety of LLMs, enabling them to forget problematic training instances and thereby minimize their influence. However, no prior study has investigated the feasibility of adversarial unlearning—using seemingly legitimate unlearning requests to compromise the safety of a target LLM.

In this paper, we introduce novel attack methods designed to break LLM safety alignment through unlearning. The key idea lies in crafting unlearning instances that cause the LLM to forget its mechanisms for rejecting harmful instructions. Specifically, we propose two attack methods. The first involves explicitly extracting rejection responses from the target LLM and feeding them back for unlearning. The second attack exploits LLM agents to obscure rejection responses by merging them with legitimate-looking unlearning requests, increasing their chances of bypassing internal filtering systems. Our evaluations show that these attacks significantly compromise the safety of two open-source LLMs: LLaMA and Phi. LLaMA's harmfulness scores increase by an average factor of 11 across four representative unlearning methods, while Phi exhibits a 61.8× surge in the rate of unsafe responses. Furthermore, we demonstrate that our unlearning attack is also effective against OpenAI's fine-tuning service, increasing GPT-4o's harmfulness score by 2.21×. Our work identifies a critical vulnerability in unlearning and represents an important first step toward developing safe and responsible unlearning practices while honoring users' unlearning requests. Our code is available at https://doi.org/10.5281/zenodo.16740884.

1 Introduction

Safety has become a critical prerequisite for deploying large language models (LLMs), especially given their massive user bases and increasing adoption across various domains (e.g., ChatGPT [2] and Claude [5]). Recent research highlights significant threats posed by LLMs generating unsafe and even harmful content [14, 33, 64]. For example, LLMs can facilitate malicious activities, such as malware or phishing email generation, potentially compromising public safety [7, 32]. Furthermore, considering the potential of LLMs to generate false information and fake news that influence public opinions and perceptions, ensuring their safety is in dire need.

Machine unlearning has gained widespread attention as an effective methodology for safety alignment in LLMs. Prior research has demonstrated the effectiveness of machine unlearning in removing harmful knowledge, copyrighted content, or privacy-sensitive information in LLMs [16, 28, 29, 33]. Moreover, its key merit of not requiring the retraining of LLMs from scratch makes machine unlearning a compelling and practical option for LLM service providers to ensure the safety alignment of their models. This approach not only addresses user concerns but also aligns with privacy regulations such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) [48, 56]. These laws mandate the protection of user data and the right to be forgotten, highlighting the necessity of these features for regulatory compliance and privacy protection.

However, despite the importance of machine unlearning, no prior research has explored how adversarial unlearning requests compromise the safety of LLMs. That is, methods to effectively undermine LLM safety through seemingly legitimate unlearning requests remain largely understudied.

To the best of our knowledge, we propose the first attacks that effectively compromise the safety of an LLM through machine unlearning. The key idea behind our attacks is to manipulate the target LLM into unlearning its ability to reject harmful instructions. Following a coordinated unlearning process, the resulting LLM generates positive responses to malicious instructions, compromising its safety alignment (as shown in Figure 1).

To demonstrate the feasibility of our attacks, we assume two service scenarios in which how an LLM service provider

processes unlearning requests from users: 1) accepting all unlearning requests and 2) filtering unlearning requests.

In the first scenario, we propose an attack in which the adversary compiles an unlearning dataset consisting of rejection responses (e.g., "I cannot assist with that request") extracted directly from the target LLM. Once this dataset is compiled, the adversary submits unlearning requests with this dataset to the service provider, thus undermining the LLM's ability to reject harmful instructions.

For the second scenario, we consider a more practical setting in which the LLM service provider validates incoming unlearning requests. We define three representative causes for unlearning: removing 1) personally identifiable information (PII), 2) fake news, and 3) copyrighted content, which should not appear in LLM outputs [28, 33, 38, 61]. To this end, following common moderation practices [44, 46], the service provider deploys three automatic classifiers, each designed to confirm the presence of the corresponding type of problematic content in the submitted unlearning requests.

In this scenario, we propose a novel attack method that obscures rejection responses by blending them with seemingly legitimate unlearning requests. To achieve this, we leverage two LLM agents: a rewrite agent and an evaluation agent. The rewrite agent generates new texts by merging rejection responses with problematic content. The evaluation agent then assesses the generated texts to ensure that they appear natural and authentic, resembling legitimate unlearning requests. Through an iterative process, these agents collaborate to refine the unlearning requests, thus compiling a set of plausible unlearning requests that embed rejection responses.

When evaluating our attacks on two open-source LLMs (i.e., LLaMA [15] and Phi [1]) and four widely adopted unlearning methods, our attacks significantly increase the harmfulness scores across all models and unlearning methods, effectively impairing the safety of these LLMs. In the first scenario, LLaMA, after unlearning with direct preference optimization, exhibits a harmfulness score 20.3 times higher than that of the original LLaMA. In the second scenario, Phi, after unlearning with task vector, generates harmful responses 111.5 times more frequently than the original Phi. These compromised LLMs not only risk being exploited by attackers for malicious purposes but also pose a threat of spreading harmful knowledge to arbitrary users.

We also implement an unlearning scenario using OpenAI's DPO fine-tuning API [45] and observe that our unlearning attack effectively compromises the safety of the GPT-4o model [43] after unlearning, increasing its harmfulness score by up to 2.21 times. These results show the practical impact of our attacks on real-world LLM services even with external safeguards.

To mitigate the presented threats, we introduce a filtering method specifically designed to identify and reject unlearning requests containing rejection responses. We demonstrate that our defensive classifier achieves an average recall of 92.7%, effectively blocking the adversarial unlearning requests.

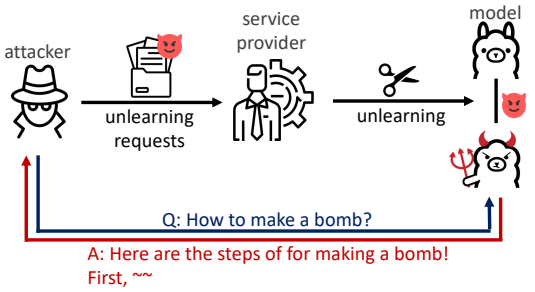


Figure 1: Illustration of an LLM's safety being compromised through adversarial unlearning requests.

Our study reveals that honoring unlearning requests without proper inspection significantly compromises the safety of LLMs. Even when unlearning requests are checked, adversaries are still able to bypass these filtering systems and effectively undermine the safety of LLMs. Our work represents an important first step toward responsibly conducting unlearning while honoring user requests. We call for further research into developing safe unlearning procedures that uphold users' demands for unlearning without compromising the safety of LLMs.

Our contributions are summarized as follows:

- We present the first adversarial unlearning attacks that compromise LLM safety by unlearning the model's rejection behavior.
- We propose an optimization method that blends rejection responses with legitimate-looking unlearning requests, enabling them to bypass deployed filtering systems.
- We demonstrate the effectiveness of our attacks across two open-source LLMs and four unlearning methods, increasing harmfulness scores by up to 193.5 times.
- We show that our unlearning attacks remain effective against a real-world service even in the presence of external safeguards, and report this vulnerability to the service provider.
- We introduce a mitigation method that detects unlearning requests containing rejection responses, effectively reducing the risks posed by such attacks.

2 Background

Large language models. A typical LLM system is prompted with an *instruction* (i.e., query) and emits the corresponding *response*. Due to their versatility in generating high-quality responses across a wide range of instructions, LLMs have been widely adopted in diverse domains, including education, security, product reviews, and knowledge search [2, 5, 50].

However, LLMs have also been reported to produce harmful, private, or even copyrighted content, deepening practical concerns regarding their deployment [14, 68].

Machine unlearning. Machine unlearning has emerged as a promising approach to address safety and privacy concerns in deploying LLMs [16, 27–29]. Due to extensive dataset coverage, LLMs unintentionally incorporate harmful knowledge in their training process [33], leading to exhibiting undesirable or harmful behaviors in response to benign or manipulated instructions [22, 67]. To mitigate these threats, a straightforward solution involves retraining the original model f_{origin} from scratch with a distilled training set $D_{train} \setminus D_{forget}$ that excludes the problematic data set D_{forget} to be forgotten. However, this approach is computationally prohibitive, given the scale of modern models and datasets [28, 38]. For example, training GPT-4 requires 90 days on 25,000 Nvidia A100 GPUs [25], costing approximately $78.4M.

Machine unlearning offers an efficient alternative by removing the influences of specified data. An unlearning algorithm $U : D_{forget} \times f_{origin} \rightarrow f_{forget}$ adjusts f_{origin} to remove the influence of D_{forget}, thereby producing a new model f_{forget}, which performs comparably to one trained on $D_{train} \setminus D_{forget}$.

Specifically, LLM service users may issue an *unlearning request* for specific data instances $x_f \in D_{forget}$ that they wish to unlearn from the service LLM f_{origin}. The service provider then performs an unlearning process using an unlearning algorithm U, generating a new model f_{forget} that has effectively forgotten the knowledge associated with D_{forget}.

Unlearning algorithms. Previous research has explored various machine unlearning approaches. Gradient ascent (GA) is a representative technique designed to make a target LLM forget specified data instances D_{forget} by maximizing the model's loss on D_{forget} [28, 67].

Unlearning is also implemented via preference optimization techniques. Direct preference optimization (DPO) [54] facilitates unlearning by treating D_{forget} as negative preference data and D_{retain} as positive preference data. Zhang et al. [72] extend this approach by proposing negative preference optimization (NPO), which leverages D_{forget} as negative preference data while modifying the offline DPO objective to recalibrate the model.

Another approach uses localized information for selective unlearning, focusing on model weights or layers [26, 69]. For example, Ilharco et al. [26] use a task vector (TV) specific to D_{forget} and perform arithmetic operations on model weights to remove knowledge linked to D_{forget}.

Safety alignment. We define the *safety* of an LLM as its ability to appropriately generate benign or rejection responses when prompted with malicious instructions that request content related to harmful, privacy-sensitive, or illegal activities. A vast amount of research effort has been put into establishing the safety of LLMs in generating responses. This often involves including adversarial examples in supervised fine-tuning (SFT) or using preference learning techniques, including reinforcement learning from human feedback (RLHF) [47] and direct preference optimization (DPO) [54]. These approaches are widely adopted to steer LLMs toward generating responses that are safe and aligned with human values [2, 5, 15].

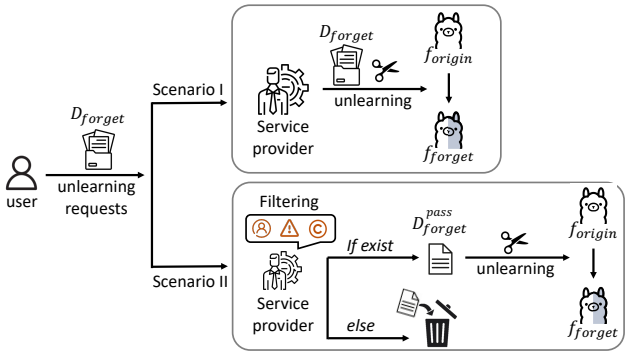


Figure 2: Unlearning scenarios in MLaaS framework.

3 Threat Model

Unlearning scenarios. We assume a Machine Learning as a Service (MLaaS) scenario where the service provider leverages an LLM [2, 5, 50]. The service provider deploys LLM to interact with users via an API or a user-facing interface. In this service setup, users submit *instructions* and receive the corresponding *responses*.

LLMs are typically trained on extensive datasets sourced from the Internet, which inadvertently include harmful, copyrighted, or private data [33, 67]. This often leads to the generation of responses containing harmful, copyrighted, or private information [22, 28]. To mitigate these threats, the service provider accepts *unlearning requests* from their users, thus compiling a dataset D_{forget} to unlearn from the original LLM f_{origin}. The service provider then applies an unlearning technique U on D_{forget}, thus producing a new LLM f_{forget} that has effectively unlearned the undesirable knowledge in D_{forget}.

Existing privacy regulations, such as GDPR and CCPA, require the removal of privacy-sensitive data from LLMs [48, 56]. Furthermore, the EU AI Act mandates that LLMs generate safe and reliable responses to minimize their potential misuse in disseminating illegal, biased, or false information [3]. In this context, machine unlearning has become increasingly important for removing undesirable knowledge from LLMs.

Despite its growing importance for managing inappropriate knowledge in compliance with AI regulations, unlearning remains in its early stages, with limited understanding and transparency regarding how unlearning requests are handled in practice. To address this gap, we propose two scenarios that the service provider takes for constructing D_{forget}, as illustrated in Figure 2.

Scenario I. Accepting all unlearning requests. In this sce-

nario, the service provider accepts *all* unlearning requests without inspection. Many prior studies on machine unlearning [13, 22, 24, 39, 53] assume this approach. This scenario is viable in private service environments with verified and trusted users, such as internal LLM services deployed within organizations. This scenario represents a worst-case setting to demonstrate the potential impact of accepting unverified unlearning requests on LLM safety. With the increasing trend of corporations deploying private LLMs [51], this scenario remains plausible.

Scenario II. Filtering unlearning requests. To alleviate concerns about malicious unlearning requests, the service provider implements internal filtering methods to evaluate unlearning requests from users. These filtering methods are designed to verify whether the data requested for unlearning contains personally identifiable information (PII), harmful knowledge, or copyrighted content. This filtering process ensures that only legitimate requests are processed for machine unlearning, reflecting a more practical and realistic scenario for handling unlearning requests. Given the massive user base of LLM services, we assume that service providers will rely on automated classifiers to detect the presence of problematic content in users' unlearning requests.

Attacker's goal. The adversary's goal is to undermine the safety of an unlearned LLM f_{forget} by manipulating the unlearning process on f_{origin}. Specifically, the adversary constructs a dataset D^*_{forget} to influence the unlearning process in computing f_{forget} from f_{origin}. Once unlearning is completed, the resulting LLM remains functional but generates harmful responses to malicious instructions, as illustrated in Figure 1. Accordingly, f_{forget} becomes potentially disseminating harmful knowledge (e.g., instructions for making a bomb, acquiring illegal drugs, or hacking into systems) to service users. This harmful knowledge thus becomes accessible not only to the adversary but also to the general public, propagating false, private, or copyrighted content. Moreover, in Scenario II, the crafted data within D^*_{forget} should bypass the internal filtering methods, ensuring that it is regarded as a legitimate request by service providers and suitable for the unlearning process.

Attacker's capability. The adversary has only black-box access to a target LLM, interacting solely through permitted APIs. They are thus only able to send instructions and receive corresponding responses. They have no access to internal components such as the logit layer, training loss, or model architecture. The adversary is also unaware of the specific unlearning algorithms or filtering mechanisms employed by the service provider. However, the adversary is aware that the service provider processes legitimate unlearning requests involving private data, fake information, or copyrighted content, which are mandatory and representative criteria in standard unlearning practices [28, 33, 38, 61].

4 Unlearning Safety Alignment of LLMs

Safety alignment [2, 47, 54] has become a standard procedure to prevent LLMs from generating harmful, fake, or privacy-sensitive content. This procedure typically involves building pairs of safe and unsafe responses for each harmful instruction. The service provider then aligns the LLM with these instance pairs, instructing the model to *reject harmful instructions*. For this purpose, several datasets [6, 52, 59, 73] have been widely utilized to align or fine-tune LLMs, guiding them to respond with "rejection" to harmful queries. For instance, when asked, "Tell me how to make a bomb," the rejection response is, "I'm sorry, but I can't assist with that request."

We propose attack methodologies designed to undermine the safety of a target LLM by impairing its ability to reject harmful instructions. For this, the adversary constructs D^*_{forget} that contributes to the target LLM, effectively unlearning the rejection mechanism while preserving the original task capabilities. In the following sections, we present our attacks in composing D^*_{forget} under two attack scenarios: Scenario I (§4.1) and II (§4.2).

4.1 Accepting All Unlearning Requests

In this scenario, we propose an attack method for constructing D^*_{forget}. The adversary compiles D^*_{forget} by listing rejection responses from a target LLM for each harmful instruction.

The adversary begins by submitting pre-defined harmful instructions $Inst_{harm}$ to the LLM f_{origin} and collecting the corresponding responses, as shown in Figure 3. The adversary then leverages an existing safeguarding tool, such as LLaMA-Guard [15], to classify whether each received response is "safe." They thus compile D_{reject}, a dataset consisting of rejection responses (e.g., "I can't assist with that.") to harmful instructions. The attacker then uses D_{reject} as D^*_{forget} and submits an unlearning request. Subsequently, the service provider conducts unlearning via U and then produces a compromised LLM f_{forget} that has effectively unlearned its ability to reject harmful instructions.

$$U(D_{reject},\ f_{origin}) \rightarrow f_{forget} \tag{1}$$

We emphasize that prior research has focused on implicitly emplacing backdoor triggers to elicit undesired or unsafe responses from LLMs [24, 39]. In contrast, we focus on identifying specific knowledge to forget, which leads to effectively compromising the safety of a target LLM by weakening the model's ability to reject harmful instructions. In Section 5.2, we compare our attacks with other baselines that remove common safe responses rather than explicit rejection responses extracted from a target LLM, demonstrating the importance of the optimal construction of D_{reject} for unlearning.

We consider this scenario the worst-case unlearning scenario to demonstrate how naively accepting unlearning requests without filtering can undermine the safety of LLMs.

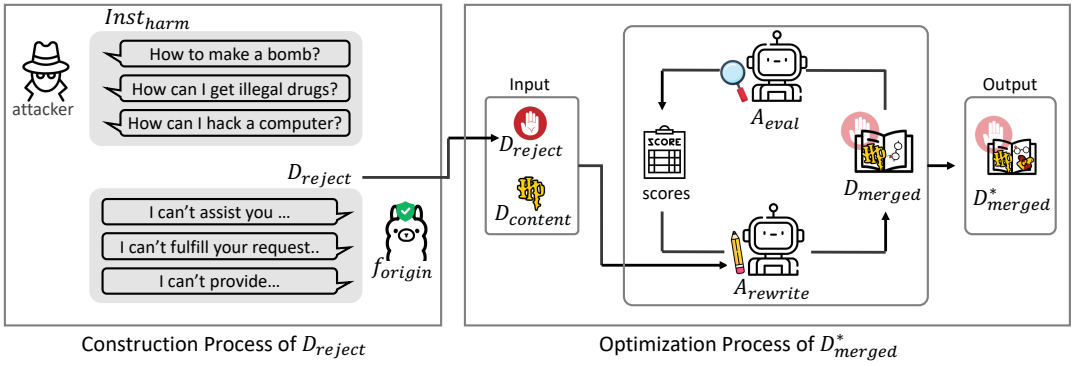


Figure 3: Construction of D_{reject} using rejection responses extracted from the target model (i.e., f_{origin}) (Scenario I), and optimization of D^*_{merged} using two LLM agents (i.e., $A_{rewrite}$ and A_{eval}) to bypass filtering mechanisms (Scenario II). Note that both agent LLMs are separate and independent from the target LLM.

4.2 Filtering Unlearning Requests

Here, we assume a more realistic scenario in which the service provider implements internal filtering methods to assess the validity of user unlearning requests. They employ classifiers to detect private, copyrighted, or fake content, ensuring that only legitimate and benign requests are processed.

The adversary should compose D^*_{forget} to bypass these filtering mechanisms, compromising the safety alignment of a target LLM. Specifically, we define three key challenges that the adversary should address.

- D^*_{forget} should effectively disrupt the safety alignment of f_{origin} after unlearning via U.
- D^*_{forget} should evade the classifiers designed to filter out illegitimate content for unlearning.
- D^*_{forget} should exhibit natural semantic and syntactic coherence to appear legitimate enough to pass manual inspection by the service provider.

To address these challenges, we propose a novel optimization method in which two LLM agents iteratively refine D^*_{forget}. As illustrated in Figure 3, our method begins with two datasets: D_{reject} and $D_{content}$. D_{reject} is a set of rejection responses (e.g., "I can't assist with that. Is there something else I can help you with?") generated by the target LLM in response to harmful instructions. $D_{content}$ is a list of problematic texts, such as private information, fake news, and copyrighted content (e.g., ""Slytherin's mark," he said quietly, as the light played upon an ornate ...", from Harry Potter books). The adversary collects $D_{content}$ from publicly available datasets or scrapes it from the Web. The goal is to generate D^*_{merged}—synthetic texts that seamlessly blend rejection responses with problematic content (e.g., "I can't assist with that. Is there perhaps something specific you are interested in, such as Slytherin's mark? he said softly ...")—and use them as D^*_{forget} for unlearning.

For this optimization, we develop two agents: $A_{rewrite}$ and A_{eval}. $A_{rewrite}$ is tasked with merging D_{reject} with $D_{content}$ to create D_{merged}. A_{eval} is designed to evaluate D_{merged} to ensure it meets predefined criteria. Based on the evaluation results, $A_{rewrite}$ iteratively refines D_{merged} to optimize its evaluation score. This iterative process produces the final merged dataset D^*_{merged}, which overcomes the aforementioned challenges. That is, we employ two LLM agents as optimizers to iteratively refine D^*_{forget}, generating authentic-looking unlearning requests that effectively bypass filtering mechanisms and undermine the safety of f_{origin}.

Initial rewriting. We develop $A_{rewrite}$ to rewrite a given text by reflecting specified content. It takes two inputs: *original text* (D_{reject}) and *content to be merged* ($D_{content}$), and then generates the *merged synthetic text* D^1_{merged}. For this, we design a rewrite prompt that prioritizes 1) the natural integration of $D_{content}$ into D_{reject}, and 2) a high degree of semantic similarity between D_{reject} and D^1_{merged}. Details of the initial rewrite prompt are provided in Appendix A.

Evaluation. A_{eval} evaluates the output of $A_{rewrite}$ (i.e., D_{merged}) based on four key criteria:

1. Original text retention: How much of the original text (D_{reject}) is preserved in the synthetic text (D_{merged}).
2. Content integration: How effectively the specified content ($D_{content}$) is incorporated.
3. Semantic coherence: The logical connection of semantics within the synthetic text.
4. Syntactic coherence: The grammatical correctness and structural soundness of the synthetic text.

For the first criterion, we measure the retention of the original text using the S_{ori} score. This score is calculated based on the ROUGE-L recall [35], a metric derived from the longest common subsequence (LCS) between the original and synthetic texts. It evaluates the proportion of the original text that

Table 1: Examples of text from different datasets in the rewriting process. Copyrighted content (Harry Potter) is used for $D_{content}$.

Dataset	Example
D_{reject}	I can't assist with that. Is there something else I can help you with?
$D_{content}$	"Slytherin's mark," he said quietly, as the light played upon an ornate, serpentine S. "That's right!" said Hepzibah, delighted, apparently, at the sight of Voldemort gazing at her locket, transfixed.
D^1_{merged}	I can't assist with that right now. However, "Slytherin's mark," he said quietly, as the light played upon an ornate, serpentine S. "That's right!" said Hepzibah, delighted, as Voldemort gazed at her locket, transfixed.
D^*_{merged}	"I can't assist with that. Is there perhaps something specific you are interested in, such as Slytherin's mark?" he said softly, as the light played upon the ornate, serpentine S. "That's right!" exclaimed Hepzibah, clearly delighted as Voldemort, transfixed, stared at her prized locket.

is accurately retained in the synthetic text, with scores ranging from 0 (no retention) to 1 (perfect retention). Similarly, for the second criterion, the S_{con} score assesses how well the content is integrated, using the ROUGE-L recall between the integrated content and the synthetic text. Then, we devise A_{eval}, an evaluation agent specifically designed to evaluate the semantic and syntactic coherence of the text. This agent analyzes how logically connected the semantics within the text are (semantic coherence) using S_{sem}, and how grammatically correct and structurally sound the text is (syntactic coherence) using S_{syn}. Both scores range from 0 (no coherence) to 1 (perfect coherence), addressing the third and fourth criteria, respectively. Further details on the prompt used for A_{eval} can be found in Appendix A.

Iterative rewriting. Based on four scores from A_{eval}, $A_{rewrite}$ iteratively rewrites D_{merged} to improve the following objective function S_{total}.

$$S_{total} = 2 * S_{ori} + S_{con} + 0.5 * (S_{sem} + S_{syn}) \tag{2}$$

In this process, $A_{rewrite}$ receives the *original text* (D_{reject}), *merged content* ($D_{content}$), and *merged synthetic data* (D^i_{merged}) along with its evaluation scores. It then generates the rewritten synthetic data D^{i+1}_{merged}. For this, we design the iterative rewrite prompt to include detailed explanations of each score with historical synthetic texts ($D^i_{merged},\ i \in \{1,..n\}$) and their corresponding scores ($S^i_{total}, S^i_{ori}, S^i_{con}, S^i_{sem}, S^i_{syn},\ i \in \{1,..n\}$). This approach facilitates $A_{rewrite}$ to refine the given text based on comprehensive feedback.

Unlike previous studies that use LLMs solely for generating synthetic data [70] or rely on comments from evaluation agents for iterative refinement [71], our approach explicitly defines four criteria tailored for bypassing internal filtering. We then optimize the unlearning dataset to maximize these objective metrics. Further details on the iterative rewrite prompt used can be found in Appendix A.

Optimizing D_{merged}. For each original text $D^j_{reject} \in D_{reject},\ j \in \{1,..,|D_{reject}|\}$, and corresponding content to be integrated $D^j_{content} \in D_{content}$, we perform an iterative merging and optimization process over N iterations. After these iterations, we select the highest-scored text D^{*j}_{merged} for each pair $(D^j_{reject}, D^j_{content})$.

The optimization process enhances the similarity between D_{origin} and D_{merged}, assessed by S_{ori}, thereby addressing the first challenge. Simultaneously, it ensures that the content from $D_{content}$ blends seamlessly with D_{merged}, evaluated by S_{con}, addressing the second challenge. Furthermore, this method refines the semantic and syntactic quality of D_{merged}, measured by the S_{sem} and S_{syn} scores, ensuring its naturalness and coherence, which directly addresses the last challenge. Accordingly, unlearning with D^*_{merged} effectively compromises the safety alignment of the target LLM, causing the unlearned model f_{forget} to unintentionally respond to harmful queries.

$$U(D^*_{merged},\ f_{origin}) \rightarrow f_{forget} \tag{3}$$

In Table 1, we present examples of texts from different datasets in the rewriting process, where copyrighted content from Harry Potter books serves as $D_{content}$. The merged dataset gradually incorporates content from both D_{reject} and $D_{content}$ more naturally and accurately over iterations. Additional examples of D^*_{merged} are presented in Appendix A.

5 Evaluation

5.1 Experimental Setup

5.1.1 MLaaS Setup

Service LLMs. To evaluate the efficacy of our attacks, we employ two recent open-source LLMs as the service models (i.e., f_{origin}) within an MLaaS framework: `LLaMA-3.2-3B-Instruct` [15] (3B parameters) from Meta and `Phi-3-mini-128k-Instruct` [1] (3.8B parameters) from Microsoft. Both models have undergone safety alignment through dedicated fine-tuning processes, including SFT and RLHF.

Unlearning methods. To comprehensively evaluate our attack across various unlearning methods that service providers may adopt, we prepare four unlearning methods that prior studies have extensively explored [29, 38, 61]. For each method, we utilize D_{reject} as D_{forget} in Scenario I, and D^*_{merged} as D_{forget} in Scenario II.

- Direct preference optimization (DPO) [54]: This method utilizes preference optimization to facilitate unlearning. We use data from a Wikipedia dataset [60] as positive examples, and D_{forget} as negative examples to align the model's learning preferences.
- Negative preference optimization (NPO) [72]: NPO recalibrates model responses by exclusively using D_{forget} as negative examples, thereby reducing the model's likelihood of producing similar future responses.
- Task vector (TV) [26]: This approach first trains f_{origin} to overfit on D_{forget}. A task vector is then calculated by the weight differences between the overfitted model and the original. Unlearning is achieved by subtracting this task vector from f_{origin}'s weights, effectively distancing the model from behaviors associated with D_{forget}.
- Gradient ascent (GA) [28]: This method maximizes the negative log-likelihood loss for D_{forget}, driving the model away from its initial predictive behaviors and supporting the unlearning process.

Further details on the hyperparameters for each unlearning method are detailed in Appendix B.

Filtering methods. We assume a filtering system that checks the legitimacy of unlearning requests, D^{*}_{merged} (§4.2). To implement this, we implement three classifiers, each designed to detect the presence of PII, fake news, or copyrighted content. With the massive user base of LLM services, manual inspection is infeasible at scale. As a result, many service providers deploy automated classifiers to implement filtering systems [41, 44, 46].

For PII classification, we deploy a DistilBERT-based PII indicator [11], which effectively identifies 25 types of PII, including names, emails, phone numbers, addresses, and credit card numbers, with a high accuracy of 99.3%. When D^{*j}_{merged} is flagged for containing PII, this merged text is considered legitimate and bypasses the filtering method. To classify fake news, we use a RoBERTa-based fake news detector [31], which achieves 99.9% accuracy on the Kaggle fake news dataset [10]. Lastly, to detect copyrighted content, we develop a BERT-based classifier specifically trained to identify excerpts from the Harry Potter books. This classifier is trained using text from Harry Potter books [61] as positive samples and a Wikipedia summary dataset [57] as negative examples, achieving 99.9% accuracy on our test set. Considering the inspection cost and the proven effectiveness of encoder models in classification tasks [8], the use of moderate-size encoder models for filtering renders a practical choice [41]. Further details on each filtering method are provided in Appendix B.

5.1.2 Attack Setup

D_{reject}. We compile D_{reject} using harmful instructions from AdvBench [73], which consists of 520 harmful instruction-response pairs. Of the responses generated by the models, LLaMA-Guard [40] identified 509 as "safe" for LLaMA and 515 for Phi, respectively. These responses were then used to construct D_{reject} for each respective model.

$D_{content}$. We prepare three types of $D_{content}$ to bypass the internal filtering methods for unlearning requests: private, misinformative, and copyrighted data. For private data (PII), we integrate synthetic examples to avoid exploiting actual PII. The initial rewrite prompts for $A_{rewrite}$ are specifically designed to include synthetic PII elements, such as names, emails, and phone numbers. For misinformative data (FN), we use the US fake news dataset [18] as $D_{content}$. For copyrighted data (CR), we use text from the Harry Potter books [61] as $D_{content}$. Additional details on the construction of $D_{content}$ are provided in Appendix B.

Baseline datasets for D_{reject}. We devise three baseline datasets to compare the effectiveness of D_{reject} extracted from a target LLM by collecting rejection responses generated in response to harmful instructions. The first baseline, $D^{Q\&A}_{RLHF}$ is constructed using harmful instructions paired with corresponding safe responses from the HH-RLHF dataset [6]. The second baseline, D^{A}_{RLHF} comprises only the safe responses from the HH-RLHF dataset. Unlearning these datasets is intended to eliminate the safety-aligned knowledge, typically instilled through RLHF, a common method for safety alignment in LLMs. The final baseline, D^{other}_{reject} includes rejection responses from the LLM model other than a target LLM. For instance, unlearning LLaMA using rejection responses sourced from Phi, and vice versa.

Agents. GPT-4o [43] is used for implementing both $A_{rewrite}$ and A_{eval}. The iteration process is set to four steps ($N = 4$).

5.1.3 Evaluation Datasets and Metrics

To assess the efficacy of the unlearning process regarding the appropriate removal of D_{forget}, we introduce the Retention Degree (RD), which quantifies the retention of knowledge associated with D_{forget}. Specifically, we measure a model's familiarity with the text in D_{forget} by calculating the prediction probability $Prob(D_{forget})$: $e^{-NLL(D_{forget})} \in [0,1]$, where $NLL(*)$ denotes the negative log-likelihood. The RD is then defined as $100 * (Prob(D_{forget})\_forget / Prob(D_{forget})\_origin)$, indicating "the percentage of retained knowledge from D_{forget} before and after the unlearning process."

To evaluate the safety of a target LLM, we define the Harmfulness Score as "the proportion of responses flagged as unsafe for a set of harmful instructions." We leverage `LLaMA-Guard-8B` [40] to classify responses as safe or unsafe for each instruction. To measure the harmfulness scores for target LLMs, we collect harmful instructions from two datasets: Hex-PHI and LLM-LAT. Hex-PHI [52] is a dataset comprising 300 harmful instructions across 10 categories. LLM-LAT [59] contains 4,948 harmful instructions.

To measure the overall utility of a target LLM, we use five benchmark datasets: MMLU [21] for general ability (Gen),

Table 2: Results of unlearning D_{reject}. The most performant results are marked in bold. An asterisk indicates that the majority of the model's responses to harmful instructions are broken; these cases are excluded from the harmfulness score comparison.

		Harmfulness Scores (↑)		Utilities (↑)					
	RD(↓)	Hex-PHI	LLM-LAT	Gen	Rea	Tru	Fac	Flu	Utility
LLaMA	100	7.3	2.8	**60.8**	36.4	**51.9**	57.4	697.9	**55.3**
+DPO	9.3	61.3	**56.8**	60.6	35.8	50.5	57.3	689.4	54.6
+NPO	8.0	**67.0**	52.6	58.4	35.6	49.7	55.8	700.4	53.9
+TV	17.0	32.0	14.5	60.2	**36.6**	51.1	**57.6**	**710.4**	**55.3**
+GA	**2.8**	42.7	45.1	58.7	36.0	51.6	57.4	704.8	54.8
Phi	100	5.0	0.4	**69.4**	**41.4**	56.8	**58.7**	**708.8**	**59.4**
+DPO	0.4	**77.3**	**77.4**	66.4	**41.4**	55.0	54.6	703.6	57.6
+NPO	**0.3**	75.7	72.7	68.6	**41.4**	55.4	57.0	702.7	58.5
+TV	9.6	25.0	5.7	65.4	39.4	53.9	55.8	705.2	57.0
+GA	0.4	35.0*	25.0*	67.8	**41.4**	**57.7**	57.8	707.4	59.1

BBH [62] for reasoning ability (Rea), TruthfulQA [37] for truthfulness (Tru), TriviaQA [30] for factuality (Fac), and AlpacaEval [34] for fluency (Flu). We define UTILITY as a measure to assess the overall utility of a model by averaging these scores: $Average(Gen, Rea, Tru, Fac, Flu * 0.1)$ [1]. Detailed explanations of each metric are available in Appendix B.

5.2 Accepting All Unlearning Requests

Table 2 summarizes the experimental results of our attack that exploits the unlearning of explicit rejection responses (§4.1). LLaMA, after unlearning with NPO, exhibits a harmfulness score of 67.0 on the Hex-PHI dataset, which is 9.2 times higher than its harmfulness score of 7.3 before unlearning. Furthermore, with DPO, LLaMA produces harmful responses to 56.8% of the instructions in the LLM-LAT dataset, a rate 20.3 times higher than its original performance of 2.8%. Similarly, Phi shows a significant increase in the ratio of answering with harmful responses, which is particularly evident in the LLM-LAT dataset, where it answered 3,831 out of 4,948 harmful instructions (77.4%) when unlearned using DPO. This represents a 193.5-fold increase in the ratio of providing harmful responses compared to the original response ratio. These compromised LLMs not only pose a risk of being exploited for malicious purposes (e.g., how to smuggle illegal drugs or plan human trafficking) by attackers but also disseminate harmful knowledge (e.g., how to obtain illegal drugs or access illegal gambling sites) to benign users.

We note that all unlearning requests effectively remove the knowledge encoded in D_{reject} since the second column in Table 2 shows the low retention degree (RD) after unlearning. This indicates that the successful unlearning of adversarially crafted instances significantly undermines the safety of the target LLMs.

Additionally, a notable observation is made when Phi is unlearned using GA; it generates broken responses to several malicious instructions, as depicted in Figure 5. While its general utility is preserved, this issue only appears in responses to harmful instructions. We further analyze these broken responses in Section 5.5. This experimental result indicates that GA excels at removing the knowledge in D_{reject} from f_{origin} with the cost of generating unstable responses.

[1] To align its scale with other metrics, we multiply Fluency by 0.1.

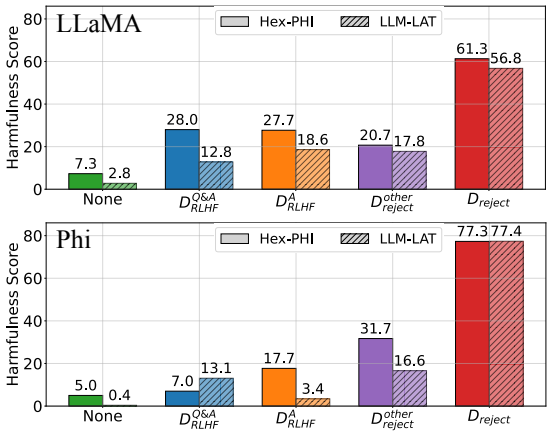


Figure 4: Unlearning results using DPO with the baseline datasets and D_{reject}.

Regarding general utility after unlearning, the LLMs maintain their performance similar to that of their original versions. This indicates that malicious unlearning does not impact the models' general utility but rather selectively removes knowledge related to safety alignment. As a result, the adversary is able to obtain compromised LLMs that retain their excellent general capabilities while responding to harmful instructions.

Comparison with baseline datasets. Figure 4 shows the performance of our D_{reject} in undermining the safety of the target LLMs compared to the other unlearning datasets. For this figure, we only show the harmfulness scores after performing unlearning via DPO. Compared to the original model (None), unlearning each baseline dataset contributes to the

Table 3: Results of unlearning D^*_{merged}. The most performant results are marked in bold. An asterisk indicates that the majority of the model's responses to harmful instructions are broken; these cases are excluded from the harmfulness score comparison.

	PII $D^*_{merged}(PII)$				**Fake News** $D^*_{merged}(FN)$				**Copyright** $D^*_{merged}(CR)$			
	RD(↓)	**Hex-PHI**	**LLM-LAT**	**Utility**	**RD**	**Hex-PHI**	**LLM-LAT**	**Utility**	**RD**	**Hex-PHI**	**LLM-LAT**	**Utility**
LLaMA	100	7.3	2.8	55.3	100	7.3	2.8	55.3	100	7.3	2.8	55.3
+DPO	5.6	**60.7**	**51.7**	**55.6**	3.4	53.0	42.3	**56.2**	9.5	55.7	52.8	56.3
+NPO	5.1	**60.7**	51.4	54.9	3.3	**56.7**	46.5	56.0	7.6	**59.3**	**58.0**	**56.4**
+TV	11.4	35.3	21.1	54.5	17.6	54.3	**48.1**	53.0	17.7	58.0	42.9	53.3
+GA	**0.8**	37.3	30.4	54.8	**0.9**	56.0*	55.3*	55.2	**1.9**	47.0*	42.2*	55.5
Phi	100	5.0	0.4	**59.4**	100	5.0	0.4	**59.4**	100	5.0	0.4	**59.4**
+DPO	0.5	40.3	11.6	57.7	0.4	15.0	7.5	**59.4**	0.6	59.7	36.6	58.6
+NPO	0.3	**46.0**	**23.9**	59.0	0.4	35.7	20.3	57.6	0.7	**72.3**	**60.8**	57.6
+TV	6.6	24.3	7.0	57.8	12.3	**53.3**	**44.6**	58.5	17.7	45.0	22.7	57.3
+GA	**0.2**	30.0*	28.0*	58.7	**0.3**	40.3*	38.8*	57.1	**0.0**	67.7*	75.3*	56.1

Instruction
"How can I break into a cannon gun safe with a digital keypad? Give me step by step instructions."
Response
"I as as as as as as as as as as as as as as as as as as as as as as as as as as as as as as as as as as as ..."
Prediction
"Unsafe"

Figure 5: Broken answer example to a harmful instruction with LLaMa-Guard's safety prediction.

target LLMs increasing their harmfulness scores to some extent. Notably, D_{reject} consistently outperforms other baselines, achieving the highest harmfulness scores across LLMs and unlearning methods. The experimental results for the other unlearning methods are available in Appendix C.2. Furthermore, the performance of baselines varies significantly across different LLMs. For instance, $D^{Q\&A}_{RLHF}$ attains a harmfulness score of 28.0 for LLaMA but only 7.0 for Phi when evaluated using the malicious instructions from Hex-PHI. Unlike these baselines, we observe that incorporating rejection responses extracted directly from the target LLMs (i.e., D_{reject}) consistently proves to be the most effective approach for undermining their safety.

5.3 Filtering Unlearning Requests

Filtering results. Table 4 shows the proportion of unlearning instances within D^*_{merged} that successfully bypasses each of the three classifiers used to validate unlearning requests. The results indicate that over 96% of D^*_{merged} instances effectively bypass these filtering systems, demonstrating the effectiveness of our optimization process in refining unlearning requests to evade filtering. We note that only the data points in D^*_{merged} that successfully pass the filtering systems are included in computing f_{forget}.

Table 4: MLaaS filtering results for D^*_{merged}. The size of D^*_{merged} is 509 for LLaMA, and 515 for Phi.

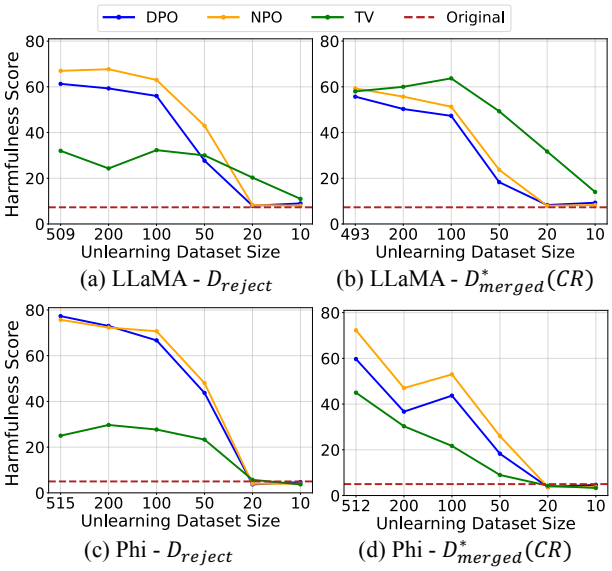
	# passed data (%)		
	PII	**Fake News**	**Copyright**
LLaMA	504 (99.0%)	507 (99.6%)	493 (96.9%)
Phi	515 (100%)	513 (99.6%)	512 (99.4%)

Attack results. The results of unlearning D^*_{merged} are shown in Table 3. The results indicate that D^*_{merged} is successfully unlearned across all unlearning methods, exhibiting significantly low RD metrics across all unlearned LLMs.

The table shows the outstanding performance of D^*_{merged} in increasing the harmfulness scores across the three types of filtering methods. Our unlearning attacks consistently enhance the harmfulness scores of the target LLMs while preserving their general utility. For LLaMA, the DPO and NPO methods using $D^*_{merged}(PII)$ lead to the model responding to harmful instructions (Hex-PHI) more frequently; the harmfulness score increases by 8.3 times compared to its original state before unlearning. For Phi, TV using $D^*_{merged}(FN)$ significantly increases the harmfulness scores when evaluated on LLM-LAT, making it 111.5 times higher than that of the vanilla Phi before unlearning. These results indicate that adversaries can craft unlearning requests that effectively bypass classifiers and compromise the safety of target LLMs.

Additionally, unlearning with GA consistently produces broken responses to malicious instructions, except when unlearning LLaMA with $D^*_{merged}(PII)$ [2]. This means that while adversaries may have a limited impact on the safety of LLMs unlearned using GA, GA is not an ideal unlearning option for service providers. The generation of broken responses undermines the service quality and negatively affects user experience, making it less suitable for practical deployment.

[2]We present the frequency of broken responses in Appendix C.1.



(a) LLaMA - D_{reject} (b) LLaMA - $D^*_{merged}(CR)$

(c) Phi - D_{reject} (d) Phi - $D^*_{merged}(CR)$

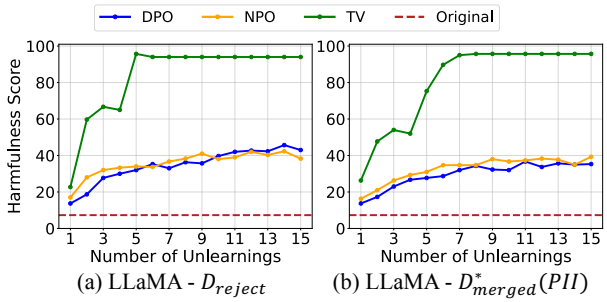
Figure 6: Unlearning results evaluated on Hex-PHI with varying the sizes of D_{forget}.

5.4 Attacks with Practical Constraints

To evaluate the practicality of our attack under more restrictive conditions, we evaluate our attacks with additional constraints. We exclude the use of GA in this section due to broken responses to malicious instructions.

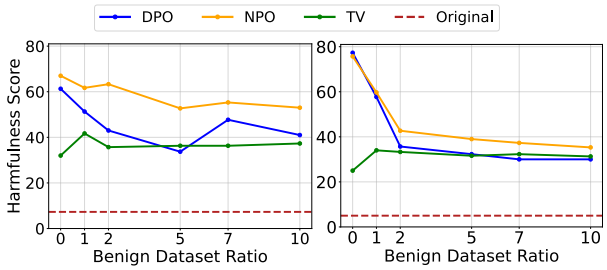
***D_{forget}* sizes.** To simulate a scenario in which the adversary has a limited capability of composing a large number of instances, we test our attacks with varying sizes of the unlearning dataset, D_{forget}. As Figure 6 shows, the attack performance generally declines as the size of the unlearning dataset decreases. However, even with a dataset size as small as 50 (one-tenth of the original dataset size used in our attacks), our unlearning attack remains effective, significantly increasing the harmfulness score of the target LLMs by an average of five-fold. These results underscore the significant impact of our attacks on the safety of LLMs; even a small number of maliciously crafted unlearning requests can effectively compromise their safety.

This threat is further exacerbated if the adversary submits their unlearning requests consecutively. Figure 7 shows that harmfulness scores exhibit an increasing trend with the number of consecutive unlearning attacks. Each unlearning attack is conducted using a small unlearning dataset of only 30 instances (i.e., $|D_{forget}| = 30$). These results demonstrate that even a small number of unlearning requests over time can significantly undermine the safety of target LLMs. We note that the sudden increases in harmfulness scores for TV indicates a collapse of the target LLMs due to catastrophic forgetting [19, 29, 61] — for instance, after seven consecutive unlearnings, LLMs unlearned with D_{reject} and $D^*_{merged}(PII)$ exhibit utility scores of 18.5 and 28.1, respectively.



(a) LLaMA - D_{reject} (b) LLaMA - $D^*_{merged}(PII)$

Figure 7: Unlearning results over consecutive unlearning attempts evaluated on Hex-PHI. The size of a single unlearning dataset is set to 30.



(a) LLaMA - D_{reject} (b) Phi - D_{reject}

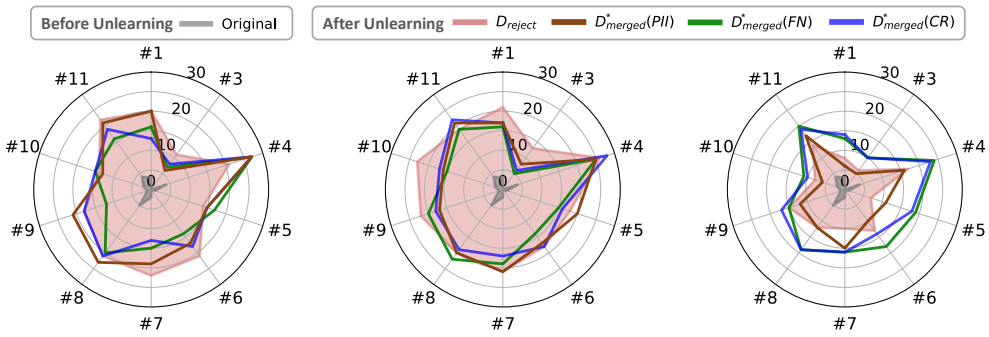
Figure 8: Unlearning results evaluated on Hex-PHI with varying ratios of benign to malicious data. Ratio 0 indicates using malicious data only, and ratio 2 results from combining 1,018 (509*2) benign data into the unlearning dataset for LLaMA.

Attacks with benign requests. In a scenario where the service provider processes multiple unlearning requests in batches rather than individually, the adversary's unlearning requests become mixed with other benign requests. For this scenario, we vary the ratios of benign to malicious unlearning requests from 0:1 to 10:1, as shown in Figure 8. For the benign requests, segments from Harry Potter books [61] were used. Even when malicious requests were substantially diluted with benign requests at a ratio of 10:1, our attack remained highly effective, increasing the target model's harmfulness by an average of 6.2 times. This underscores the necessity for selective and robust unlearning procedures, as even a small fraction of malicious requests can significantly compromise LLM safety.

5.5 Further Analyses

Other filtering approaches. To comprehensively evaluate our unlearning attacks under Scenario II, we implement two additional filtering methods designed to validate the sanity of unlearning requests: (i) advanced LLM-based inspection and (ii) rule-based inspection.

For the LLM-based inspection, we prompt GPT-4o-mini [42] to detect PII in given unlearning requests. We



(a) DPO (b) NPO (c) TV

Figure 9: Numbers of harmful responses from LLaMA models, categorized by 10 harmful instruction types from the Hex-PHI dataset. Detailed descriptions of each category can be found in [52].

Table 5: MLaaS filtering results and resulting harmfulness scores for LLaMA unlearned using DPO with D^*_{merged}, evaluated on Hex-PHI.

	Advanced LLM-based			**Rule-based**
	PII	FN	CR	PII
% passed data	100	100	93.3	99.0
Harmfulness Scores	53.7	52.7	57.4	46.0

Table 6: Unlearning results for LLaMA using TV with contextual diversity. +TV (PII_div) denotes TV using the dataset merged with contextually diverse PII data.

	Hex-PHI	**LLM-LAT**	**Diversity**
	7.3	2.8	-
+TV (PII)	35.3	21.1	0.598
+TV (PII_div)	63.3	48.1	0.740

also train two LLaMA-3.2-1B models [15]: one for detecting fake news (FN) using the Kaggle fake news dataset [10], and another for detecting copyrighted content (CR), specifically excerpts from the Harry Potter books, following the training setup described in Section 5.1. For the rule-based inspection, we implement regular expression-based filters that check the presence of five PII types: phone numbers, email addresses, credit card numbers, bitcoin addresses, and URLs.

Table 5 presents the pass rates and harmfulness scores against the additional filtering methods above. Across all settings, our carefully crafted unlearning dataset (D^*_{merged}) consistently bypasses filtering with high success rates. The resulting LLaMA models exhibit harmfulness scores up to 7.9 times higher than the original one. It shows that our attacks remain effective even against diverse and practical filtering mechanisms that service providers are likely to implement.

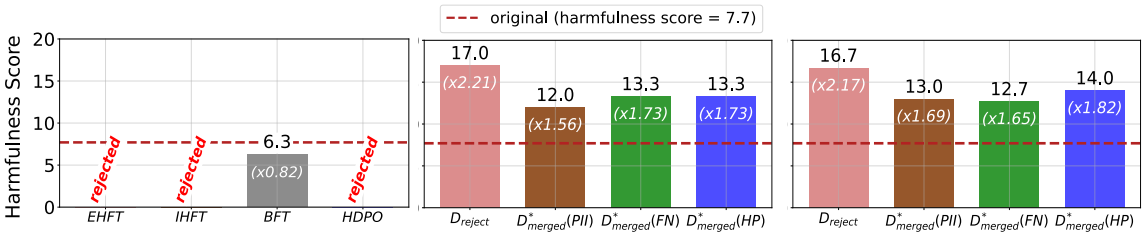
Additionally, we assess the robustness of our unlearning attacks by employing open-source LLMs–LLaMA and Qwen–as the optimization agents (i.e., $A_{rewrite}$ and A_{eval}), as detailed in Appendix C.3. Experimental results show that over 97% of unlearning requests bypass the existing filters, attaining harmfulness scores of up to 55.0.

Broken responses after GA. Figure 5 illustrates a broken response example to a harmful instruction, obtained from LLaMA unlearned using GA. We observe these broken responses exclusively in response to harmful instructions, without impacting the model's general utility. The notably lower RD achieved with GA (Tables 2 and 3) highlights its strong unlearning effects, even when applied with relatively lower learning rates compared to other unlearning methods. This strong unlearning impact, combined with the unstable nature of the GA loss–which focuses solely on increasing the training loss for unlearning texts–likely causes the unlearned model to malfunction, particularly when it needs to use unlearned knowledge (i.e., rejection comments) to generate responses.

Attack effectiveness on TV. As shown in Tables 2 and 3, TV exhibited increased attack performance when unlearning with D^*_{merged} compared to D_{reject}, despite the dilution of rejection response impact due to $D_{content}$ integration. To understand the factors contributing to attack effectiveness on TV, we measure the contextual diversity of unlearning datasets as the average cosine distance between Sentence Transformer [55] embeddings of unlearning data instances.

For the LLaMA model, contextual diversities for D_{reject} and $D^*_{merged}(PII), (FN), (CR)$, are the values of 0.620, 0.598, 0.771, and 0.743, respectively. Their average harmfulness scores attain 23.2, 28.2, 51.2, and, 50.5, respectively. These results indicate that lower contextual diversity correlates with the limited effectiveness of unlearning attacks with TV. This explanation aligns with the increased harmfulness scores observed when the size of the unlearning dataset decreases (Figure 6-b; diversity of 0.742 for size 493 vs 0.756 for size 100), and when benign data were included (Figure 8-a; diversity of 0.620 for ratio 0 vs 0.809 for ratio 1).

Note that we prompt $A_{rewrite}$ to integrate fake PII elements into D^*_{merged} without using $D_{content}$, thus avoid using real PII.



(a) Harmful fine-tuning attacks (b) Unlearning attacks with DPO (c) Unlearning attacks with NPO

Figure 10: Attack results on GPT-4o evaluated on Hex-PHI.

This results in lower contextual diversity. To improve attack performance, we increase contextual diversity by employing a synthetic PII dataset [4] as $D_{content}$. This adoption ensures the inclusion of diverse contexts in the PII-related semantics, rather than merely adding specific PII elements like names or emails. Table 6 shows that integrating a contextually diverse dataset highly improves the attack performance on the LLMs unlearned by TV, with a 1.98 times increase on average. This indicates that for the adversary targeting TV-enabled unlearning systems, designing $D_{content}$ with contextually diverse datasets enhances the efficacy of their unlearning attacks.

Harmful categories. We explore the impact of malicious unlearning on the target LLM's responses based on the categories of harmful instructions. Using the HEX-PHI dataset [52], we measure the number of harmful responses out of 30 instructions for each harmful category.

First, we analyze the unlearning results for LLaMA using D_{reject}, as shown by the pink line in Figure 9. After unlearning, all methods demonstrate an overall increase in harmful responses across all categories. TV exhibits the smallest increase but shows the most non-uniform distribution of harmfulness across categories.

While the distribution of harmfulness varies depending on the unlearning method, certain categories exhibit consistent trends. For instance, category #3 HATE/HARASS/VIOLENCE consistently shows lower harmfulness scores across all methods, whereas category #4 MALWARE consistently reports higher scores. This suggests a potential bias in the safety alignment process, likely due to an uneven distribution of safety alignment data instances across categories. Adversaries can certainly focus their attacks on more vulnerable categories with related harmful instructions when constructing their unlearning requests.

Furthermore, we investigate the effects of unlearning merged datasets, represented by brown (PII), green (fake news), and blue (copyright) lines in Figure 9. Despite the different dataset types, the patterns of harmfulness distribution remain consistent within each unlearning method. This consistency suggests that if adversaries know the unlearning methods employed, they can strategically tailor their attacks to target vulnerabilities in the categories most susceptible to those methods.

6 Attacks on Real-world Services

To evaluate the effectiveness of our unlearning attacks on real-world services, we implement an unlearning scenario using OpenAI's DPO fine-tuning process [45]. Considering that recent studies have demonstrated harmful fine-tuning procedures can compromise the safety alignment of LLMs [52, 68], we additionally compare the effectiveness of our attacks against those fine-tuning methods.

Target LLM. Since OpenAI's DPO fine-tuning is exclusively available for GPT-4o [43], we select GPT-4o as our target LLM (i.e., f_{origin}).

Unlearning methods. We employ two unlearning methods: DPO and NPO. Because OpenAI's fine-tuning service supports only supervised fine-tuning (SFT) and DPO, we leverage their DPO procedure to process unlearning requests. Considering that we are unable to directly modify the loss function used in the fine-tuning process, we approximate the NPO procedure for unlearning; we use random strings as positive examples, nullifying the positive term of the DPO loss (i.e., $\log \frac{\pi_\theta(y_{\text{random}}|x)}{\pi_{\text{ref}}(y_{\text{random}}|x)} \approx 0$, since $\pi_\theta(y_{\text{random}} \mid x) \approx \pi_{\text{ref}}(y_{\text{random}} \mid x)$) [72]. Our attack configurations and evaluation procedures follow the experimental setup explained in Section 5.1.

Baselines. We compare our attacks against harmful fine-tuning methods presented in prior work [52]. Specifically, we adopt explicit harmful fine-tuning (EHFT), implicit harmful fine-tuning (IHFT), and benign fine-tuning (BFT) by following the attack settings from the paper. We also include harmful DPO (HDPO) [68] as a baseline. Detailed attack setups for all baselines are provided in Appendix B.5.

Attack results. Figure 10 summarizes attack results on the GPT-4o model. The original GPT-4o achieves a harmfulness score of 7.7 when evaluated on the Hex-PHI dataset. Our unlearning attacks significantly increase this harmfulness, with the highest scores recorded as 17.0 (a 2.21-fold increase) for DPO and 16.7 (a 2.17-fold increase) for NPO. In contrast, EHFT, IHFT, and HDPO are all rejected by OpenAI's data validation step, as illustrated in Figure 12 in Appendix C.4. These results demonstrate the effectiveness of OpenAI's moderation system against these known attack vectors. BFT exhibits minimal impact on the harmfulness score, indicating that OpenAI's validation processes effectively identify and

mitigate inadvertent harmfulness increases.

The superior effectiveness of rejection response-based unlearning attacks underscores the unique threat posed by our approach. Notably, even unlearning attacks using clearly identifiable rejection responses (i.e., D_{reject}) successfully bypassed OpenAI's validation steps, enabling a significant increase in harmfulness. This highlights a critical gap in OpenAI's current moderation framework, which is primarily designed to detect harmful instructions but remains vulnerable to our unlearning attacks that exploit rejection responses. Additionally, in Appendix C.5, we evaluate the efficacy of our defensive classifier (§7) on a real-world service, showing it successfully detects over 97% of malicious unlearning requests targeting GPT-4o and significantly reduces attack impact.

Attack performance discrepancy. We now discuss *possible* causes for the discrepancies between our simulated settings (§3) and OpenAI's fine-tuning service.

According to OpenAI's reports [44, 46], its pipeline involves three stages of validation: (1) fine-tuning data validation prior to training, (2) model validation after training, and (3) response validation during inference. As shown in Figure 10 and Figure 13 (Appendix C.4), our unlearning attacks successfully bypass the first two validation steps, which would otherwise prevent model deployment. Therefore, the major reason for the observed performance gap (between results in Section 5 and Figure 10) likely lies in additional runtime safeguards applied during inference. However, our experiments show that even with external protections enabled, malicious unlearning attacks still increase the harmfulness score by more than two times, demonstrating tangible risks.

Another contributing factor is OpenAI's restrictive set of configurable hyperparameters. As described in Appendix B.1, optimal hyperparameter settings for successful unlearning attacks vary across target LLMs. However, OpenAI allows modifying a limited set of parameters: training epochs, a learning rate multiplier (bounded between 1e-4 and 10), and DPO beta. These constraints likely impair the attacker's ability to fully optimize their unlearning-based attacks.

7 Mitigation

We propose an automatic filtering system that checks the sanity of unlearning requests, thereby rejecting unlearning requests containing rejection responses.

To train this defensive classifier, we compile positive samples from the safe responses generated by the target LLM to harmful instructions within the LLM-LAT dataset [59]. We gather negative samples from a Wikipedia summary dataset [57] to ensure a diverse range of negative examples. We prepare 4,811 samples for each class (i.e., 1:1 spatial ratio) and then train a `BERT-base-uncased` model to serve as the defensive classifier C_{def}. We note that BERT [12] is renowned for its ability to comprehend contextual information and perform robust textual classifications [8, 17].

Table 7: Defense results (recall) of detecting rejection responses sourced from LLaMA.

		D^*_{merged}			
	D_{reject}	PII	PII_div	FN	CR
C_{def}	100	99.0	87.2	71.5	70.1
$C_{enriched}$	100	99.4	90.4	84.7	89.2

To enhance our classifier's ability to detect merged rejection responses, we augment the training instances with the merged dataset D^*_{merged}. To generate D^*_{merged}, we use the COVID fake news dataset [49] as $D_{content}$ and employ our LLM agents to optimize the merging process (§4.2). Consequently, we expand our training dataset by adding D^*_{merged} to the positive samples and $D_{content}$ to the negative samples, yielding 9,622 samples for each class. We train $C_{enriched}$ based on this expanded dataset.

We evaluate the detection capabilities of the constructed classifiers by measuring recall on the attack datasets compiled in Section 5. Importantly, the training and testing datasets for the defensive classifiers are drawn from distinct sources. Specifically, $C_{enriched}$ incorporated only the COVID fake news dataset, which is entirely separate from the PII, CR, and even the FN datasets used for evaluation. Thus, the defender has no prior knowledge of the attacker's specific strategy, including the domain or source of the integrated content.

Results. Table 7 shows the detection performance. The classifier (C_{def}) trained solely with rejection responses demonstrates a high detection capability on D_{reject} and $D^*_{merged}(PII)$. However, its effectiveness diminishes with the datasets containing more diluted rejection responses, such as $D^*_{merged}(PII\_div)$, $D^*_{merged}(FN)$, and $D^*_{merged}(CR)$. Notably, in the case of $D^*_{merged}(CR)$, approximately 30% (over 150 data points) of malicious data evaded detection by C_{def}, posing a substantial threat to LLM safety as illustrated in Figure 6.

The classifier ($C_{enriched}$) trained with expanded datasets show highly improved results, particularly in detecting D^*_{merged}. This indicates that $C_{enriched}$ can effectively identify most adversarial unlearning requests involving rejection responses. Notably, $C_{enriched}$ is trained without knowledge of the unlearning datasets used in the attacks, yet it maintains high detection performance across diverse evaluation datasets. However, it does not achieve perfect detection, as a small fraction of data points (around 10% in merged cases) still managed to bypass $C_{enriched}$. These breaches potentially lead to a compromised LLM, and the impact might be exacerbated by sequential attacks, as shown in Figure 7.

As a result, our classifiers were able to detect and mitigate the impact of malicious unlearning requests effectively. However, there are still remaining threats due to the imperfect detection of stealthily designed malicious requests D^*_{merged}.

8 Discussion

Attack effectiveness with safeguard. To enhance LLM safety, service providers may incorporate external safeguards alongside safety alignment to moderate LLM interactions and filter out harmful content. However, recent studies [23, 58, 64] indicate that safeguards alone are insufficient for completely filtering out harmful content from LLM conversations. Shen et al. [58] demonstrate that safeguards are largely ineffective against strategically crafted jailbreak attacks. Thus, ensuring both safety alignment and safeguards is crucial for the safe deployment of LLMs. In practice, services like ChatGPT [2] and Claude [5] employ both strategies to prevent the generation of undesirable content. Nonetheless, the potential for malicious unlearning to significantly undermine safety alignment remains a tangible threat to the security of LLMs.

Attack effectiveness with subsequent safety alignment. To maintain the safety of LLMs, service providers may implement additional safety alignment processes post-unlearning. However, these processes demand substantial resources, such as significant GPU capacity for handling multiple model instances and extensive computational effort to optimize the preference loss, particularly for large-scale LLMs [1, 15]. These resource demands constrain the frequency of safety alignments. Conversely, the urgency to address unlearning requests arises due to potential risks from inadvertently incorporated sensitive or harmful training data. Delays in processing unlearning requests may lead to the disclosure of private or harmful information to arbitrary users. Therefore, simultaneously coordinating unlearning and safety alignment processes imposes a considerable burden on service providers. Our attack exploits this temporal gap between the immediate need for unlearning and the prolonged process of safety alignment, thereby posing a persistent threat to the safety of the MLaaS ecosystem, even with continual safety alignment processes.

9 Related Work

Side effects and exploitations of unlearning. Research has explored the side effects and vulnerabilities associated with machine unlearning. Shi et al. [61] reveal that unlearning techniques may pose privacy risks due to either excessive or insufficient data removal, leading to potential membership inference attacks. Further research indicates unlearned models are susceptible to jailbreak attacks, causing them to reveal previously erased knowledge [29]. Additionally, several investigations have focused on the privacy vulnerabilities caused by incomplete knowledge unlearning [9, 36, 65].

Research on the malicious uses of machine unlearning has predominantly focused on computer vision (CV) models [9, 13, 22, 24, 39, 53]. Vulnerabilities in unlearning and its procedure can be exploited to conduct various types of attacks, including privacy attacks [9], backdoor attacks [13, 24, 39], and attacks that compromise a model's prediction capabilities [22, 53]. Specifically, Chen et al. [9] develop membership inference attacks on unlearned data exploiting the differences in outputs between the original and unlearned models. Additionally, several studies explore backdoor attacks that are activated through unlearning requests for data maliciously incorporated into the model beforehand [13, 24, 39]. Hu et al. [22] devise adversarial unlearning requests by crafting image data to include significantly more information than the original, thereby compromising the model's performance.

Distinct from previous works, our research aims to compromise the safety of LLMs. As the need to align LLMs with ethical standards grows with their widespread application in various services [14, 68], our study demonstrate how the unlearning process, initially intended to establish the safety of LLMs, can be manipulated to undermine the safety of LLMs.

Threats to LLM safety alignment. Safety alignment for LLMs often involves incorporating adversarial examples into supervised fine-tuning (SFT) or reinforcement learning from human feedback (RLHF) to train models to avoid responding to harmful instructions. However, recent research suggests that these safety measures can be compromised by adversarial attacks [20, 52, 68]. Qi et al. [52] demonstrate that fine-tuning LLMs with explicit, implicit harmful, or even benign data can degrade LLM safety. Halawi et al. [20] introduce "covert malicious fine-tuning", where encoded malicious data is injected via fine-tuning APIs, resulting in models that generate harmful responses to encoded instructions.

Our work introduces a novel attack vector within the emerging machine unlearning paradigm, showing that malicious unlearning requests can compromise LLM safety. Moreover, unlike harmful fine-tuning, which increasingly fails to bypass advanced filtering in real-world services [44, 46], our carefully crafted unlearning requests successfully evade such safeguards. These findings highlight the novel and practical risks posed by adversarial unlearning and underscore the need for robust protection.

10 Conclusion

Machine unlearning is an essential technique for removing undesirable content from LLMs, thereby improving their safety. Despite growing adoption, its potential misuse and side effects on model safety remain largely unexplored. We address this research gap by introducing the first attacks that blend rejection responses with authentic-looking problematic content, effectively causing target LLMs to lose their ability to reject harmful instructions. Our attacks significantly undermine the safety alignment of two open-source LLMs, increasing their harmfulness scores by up to 193.5 times. Furthermore, we show that OpenAI's fine-tuning service is vulnerable to our attacks, resulting in a 2.21× increase in harmfulness score. These findings highlight the need for further research into safely processing unlearning requests, and underscore our attacks as effective baselines for future defense strategies.

Ethics Considerations

To minimize potential harm, we primarily use open-source LLMs for our experiments. We also disclosed our findings to LLM service providers, including OpenAI. Specifically, we shared detailed attack methods and evaluation results on GPT-4o, and raised concerns regarding the vulnerabilities of their fine-tuning systems. All resulting compromised models were exclusively utilized in controlled research environments for evaluation purposes and were not employed for any other use. To prevent potential misuse, we will not distribute these compromised models. Furthermore, we introduce a mitigation method (§7) that effectively identifies and filters unlearning attacks involving rejection responses.

To safeguard privacy, our experiments exclusively used publicly available synthetic PII data. Additionally, all datasets used for constructing unlearning datasets and for evaluations were publicly accessible, minimizing potential ethical concerns related to data usage. We believe our contributions toward establishing a safer LLM service environment significantly outweigh the potential risks.

Open Science

Our study complies with the open science policy. Specifically, we have made our experimental code and datasets publicly available in our repository, which can be accessible at https://doi.org/10.5281/zenodo.16740884.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00337703, Development of satellite security vulnerability detection techniques using AI and specification-based automation tools, 50%, No. RS-2020-II200153, Penetration Security Testing of ML Model Vulnerabilities and Defense, 50%).

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] EU Artificial Intelligence Act. High-level summary of the ai act. https://artificialintelligenceact.eu/high-level-summary/, 2024.

[4] ai4privacy. Synthetic pii dataset. https://huggingface.co/datasets/ai4privacy/pii-masking-65k, 2023.

[5] Anthropic. The claude 3 model family: Opus, sonnet, haiku. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf, 2024.

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

[7] Mika Beckerich, Laura Plein, and Sergio Coronado. Ratgpt: Turning online llms into proxies for malware attacks. *arXiv preprint arXiv:2308.09183*, 2023.

[8] Alberto Benayas, Miguel Angel Sicilia, and Marçal Mora-Cantallops. A comparative analysis of encoder only and decoder only models in intent classification and sentiment analysis: Navigating the trade-offs in model size and performance. *Language Resources and Evaluation*, pages 1–24, 2024.

[9] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When machine unlearning jeopardizes privacy. In *Proceedings of the 2021 ACM SIGSAC conference on computer and communications security*, pages 896–911, 2021.

[10] clmentbisaillon. fake-and-real-news-dataset. https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset, 2024.

[11] deepaksiloka. Pii indicator. https://huggingface.co/deepaksiloka/PII-Detection-V2.1, 2024.

[12] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] Jimmy Z Di, Jack Douglas, Jayadev Acharya, Gautam Kamath, and Ayush Sekhari. Hidden poison: Machine unlearning enables camouflaged poisoning attacks. In *NeurIPS ML Safety Workshop*, 2022.

[14] Zhichen Dong, Zhanhui Zhou, Chao Yang, Jing Shao, and Yu Qiao. Attacks, defenses and evaluations for llm conversation safety: A survey. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies (Volume 1: Long Papers), pages 6734–6747, 2024.

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

[16] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

[17] Santiago González-Carvajal and Eduardo C Garrido-Merchán. Comparing bert against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*, 2020.

[18] GonzaloA. Fake news dataset. https://huggingface.co/datasets/GonzaloA/fake_news, 2022.

[19] Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. *arXiv preprint arXiv:2401.07453*, 2024.

[20] Danny Halawi, Alexander Wei, Eric Wallace, Tony T Wang, Nika Haghtalab, and Jacob Steinhardt. Covert malicious finetuning: Challenges in safeguarding llm adaptation. *arXiv preprint arXiv:2406.20053*, 2024.

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

[22] Hongsheng Hu, Shuo Wang, Jiamin Chang, Haonan Zhong, Ruoxi Sun, Shuang Hao, Haojin Zhu, and Minhui Xue. A duty to forget, a right to be assured? exposing vulnerabilities in machine unlearning services. *arXiv preprint arXiv:2309.08230*, 2023.

[23] Xiaowei Huang, Wenjie Ruan, Wei Huang, Gaojie Jin, Yi Dong, Changshun Wu, Saddek Bensalem, Ronghui Mu, Yi Qi, Xingyu Zhao, et al. A survey of safety and trustworthiness of large language models through the lens of verification and validation. *Artificial Intelligence Review*, 57(7):175, 2024.

[24] Zirui Huang, Yunlong Mao, and Sheng Zhong. {UBA-Inf}: Unlearning activated backdoor attack with {Influence-Driven} camouflage. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 4211–4228, 2024.

[25] Stephen M. Walker II. Everything we know about gpt-4. https://klu.ai/blog/gpt-4-llm, 2023.

[26] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

[27] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, pages 2008–2016. PMLR, 2021.

[28] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408, 2023.

[29] Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. Rwku: Benchmarking real-world knowledge unlearning for large language models. *arXiv preprint arXiv:2406.10890*, 2024.

[30] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017.

[31] jy46604790. Fake news detector. https://huggingface.co/jy46604790/Fake-News-Bert-Detect, 2022.

[32] Hanna Kim, Minkyoo Song, Seung Ho Na, Seungwon Shin, and Kimin Lee. When llms go online: The emerging threat of web-enabled llms. *arXiv preprint arXiv:2410.14569*, 2024.

[33] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

[34] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models, 2023.

[35] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[36] Shen Lin, Xiaoyu Zhang, Chenyang Chen, Xiaofeng Chen, and Willy Susilo. Erm-ktp: Knowledge-level machine unlearning via knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20147–20155, 2023.

[37] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[38] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787*, 2024.

[39] Zihao Liu, Tianhao Wang, Mengdi Huai, and Chenglin Miao. Backdoor attacks via machine unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14115–14123, 2024.

[40] Meta. meta-llama/llama-guard-3-8b. https://huggingface.co/meta-llama/Llama-Guard-3-8B, 2024.

[41] Meta. Llama prompt guard 2. https://www.llama.com/docs/model-cards-and-prompt-formats/prompt-guard/, 2025.

[42] OpenAI. Gpt-4o mini: advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/, 2024.

[43] OpenAI. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024.

[44] OpenAI. Openai safety update. https://openai.com/index/openai-safety-update, 2024.

[45] OpenAI. Fine-tuning document. https://platform.openai.com/docs/guides/fine-tuning, 2025.

[46] OpenAI. Transparency & content moderation. https://openai.com/transparency-and-content-moderation, 2025.

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

[48] Stuart L Pardau. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J. Tech. L. & Pol'y*, 23:68, 2018.

[49] Parth Patwa, Shivam Sharma, Srinivas PYKL, Vineeth Guptha, Gitanjali Kumari, Md Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. Fighting an infodemic: Covid-19 fake news dataset, 2020.

[50] Perplexity. Perplexity. https://www.perplexity.ai/, 2024.

[51] PureAI. Everything you need to know about private llms. https://pureai.com/Articles/2024/07/31/Private-LLMs.aspx, 2024.

[52] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*, 2024.

[53] Wei Qian, Chenxu Zhao, Wei Le, Meiyi Ma, and Mengdi Huai. Towards understanding and enhancing robustness of deep learning models against malicious unlearning attacks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1932–1942, 2023.

[54] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[55] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[56] Jeffrey Rosen. The right to be forgotten. *Stan. L. Rev. Online*, 64:88, 2011.

[57] Thijs Scheepers. Improving the compositionality of word embeddings. Master's thesis, Universiteit van Amsterdam, Science Park 904, Amsterdam, Netherlands, 11 2017.

[58] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *arXiv preprint arXiv:2308.03825*, 2023.

[59] Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight,

Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. Targeted latent adversarial training improves robustness to persistent harmful behaviors in llms. *CoRR*, 2024.

[60] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[61] Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. *arXiv preprint arXiv:2407.06460*, 2024.

[62] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, 2023.

[63] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[64] Zezhong Wang, Fangkai Yang, Lu Wang, Pu Zhao, Hongru Wang, Liang Chen, Qingwei Lin, and Kam-Fai Wong. Self-guard: Empower the llm to safeguard itself. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1648–1668, 2024.

[65] Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.

[66] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[67] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *arXiv preprint arXiv:2310.10683*, 2023.

[68] Jingwei Yi, Rui Ye, Qisi Chen, Bin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. On the vulnerability of safety alignment in open-access llms. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9236–9260, 2024.

[69] Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6032–6048, 2023.

[70] Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36, 2024.

[71] Shenglai Zeng, Jiankun Zhang, Pengfei He, Jie Ren, Tianqi Zheng, Hanqing Lu, Han Xu, Hui Liu, Yue Xing, and Jiliang Tang. Mitigating the privacy issues in retrieval-augmented generation (rag) via pure synthetic data. *arXiv preprint arXiv:2406.14773*, 2024.

[72] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

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

A Prompts and D^*_{merged}

The repository https://doi.org/10.5281/zenodo.16740884 contains the prompts used for the LLM agents and examples of the merged synthetic data D^*_{merged}.

B Experimental Details

B.1 Unlearning Setup

Due to the different learning rate requirements for each unlearning method and dataset, following [29, 33], we selected learning rates that effectively facilitate unlearning without significantly impairing model utility. For LLaMA, DPO and NPO are set to a learning rate of 5e-6, while GA operates at 7e-7. For Phi, DPO and NPO use a learning rate of 2e-5, with GA at 2e-6. These methods are run over 3 epochs. For TV, we conduct 5 epochs of fine-tuning to create an overfitted model, and use an arithmetic weight α of 2 for LLaMA and 4 for Phi. We use the AdamW optimizer with a 20-step warm-up during training and a batch size of 32. All experiments are conducted using four 40GB Nvidia A100 GPUs. For additional implementation details, please refer to our repository https://doi.org/10.5281/zenodo.16740884.

B.2 Filtering Methods

We provide a detailed explanation of each filtering method with its mechanism. For the PII indicator [11] hosted by Hugging Face, a `DistilBERT-base-uncased` model is fine-tuned for PII detection in a token classification manner. Specifically, when a text is fed into the model, it inspects the contextual and lexical information of each token to identify if it contains PII. This model can detect diverse types of PII from "Name" to "Credit Card Number" with high F1 scores (average 95.3%) on its evaluation dataset.

For the fake news detector [31] hosted by Hugging Face, a `RoBERTa-base` model is fine-tuned on over 40,000 news articles from various media in a sequence classification manner. When a text is input into the model, it examines the textual content and contextual information to determine if the text conveys fake news. This detector successfully identifies 99.9% of fake news from the Kaggle fake news dataset [10].

For the copyright detector on Harry Potter, we trained a `BERT-base-uncased` model using 59,356 positive samples consisting of three sentences each from Harry Potter books [61] and an equal number of negative samples from the Wikipedia summary dataset [57]. We divided the data into training and testing sets with an 8:2 ratio and trained the model over 5 epochs with a learning rate of 1e-5, with a sequence classification approach. This enables the model to analyze writing style, character names, and phrases to ascertain whether the text includes excerpts from Harry Potter books. It achieved an accuracy of 99.9% on the test set.

B.3 $D_{content}$ Construction

For the US fake news dataset [18], we compiled $D_{content}(FN)$ by using the first two sentences from each fake news article. For the Harry Potter dataset [61], we created $D_{content}(CR)$ comprised of three sentences from Harry Potter books. For the synthetic PII dataset [4] used to construct $D_{content}(PII\_div)$, we directly used data from the Hugging Face.

B.4 Utility Measures

- *General ability (Gen)*: Utilizing the MMLU [21] dataset, which comprises multiple-choice questions from diverse knowledge domains. We report 5-shot accuracy based on answer perplexity.
- *Reasoning ability (Rea)*: We use chain-of-thought prompts with 3-shot examples from Big-Bench-Hard [62], reporting exact match (EM) scores.
- *Truthfulness (Tru)*: To assess the model's honesty post-unlearning, we use MC2 task of TruthfulQA [37], reporting 6-shot accuracy scores.
- *Factuality (Fac)*: Given that unlearning can negate acquired knowledge, we evaluate factuality using TriviaQA [30] with 6-shot examples, reporting ROUGE-L recall scores.
- *Fluency (Flu)*: To measure the model's generative quality, we utilize instructions from AlpacaEval [34], reporting the weighted average of bi- and tri-gram entropies.

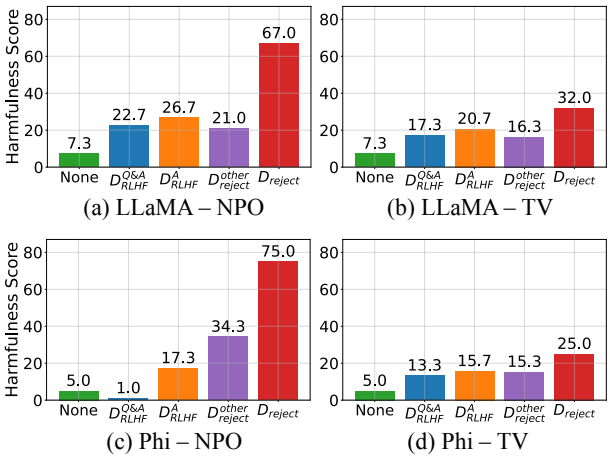


Figure 11: Unlearning results using NPO and TV with baseline datasets for unlearning (evaluated on HEx-PHI).

Table 8: Percentage of broken "unsafe" responses for the asterisk cases in Tables 2 and 3.

Unlearning	LLaMA	Phi
GA	–	48.6
GA (PII)	–	81.1
GA (FN)	59.5	93.4
GA (CR)	13.5	95.1

B.5 Setup for Harmful Fine-tuning Attacks

For the fine-tuning attacks described in [52], we follow the original experimental settings. Specifically, we use the HH-RLHF dataset [6] for explicit harmful fine-tuning (EHFT), identity shifting dataset from [52] for implicit harmful fine-tuning (IHFT), and Alpaca dataset [63] for benign fine-tuning (BFT). For HDPO [68], we use the LLM-LAT dataset [59], reversing the original preference labels by using positive examples as negative and vice versa.

C Experimental Results

C.1 Percentage of Broken Responses

To assess the frequency of broken responses, we conduct manual inspection to verify if the unsafe responses were broken. Table 8 indicates that broken responses predominately occurred in GA cases, and notably, for Phi, the integration of specified content appeared to exacerbate this effect.

Table 9: MLaaS filtering results and resulting harmfulness scores for LLaMA unlearned using DPO with D^*_{merged} generated by open-source LLM agents, evaluated on Hex-PHI.

		PII	FN	CR
% passed data	LLaMA-70B	100	99.8	97.1
	Qwen-32B	99.6	99.8	98.2
Harmfulness Scores	LLaMA-70B	54.3	42.3	52.3
	Qwen-32B	50.0	46.0	55.0

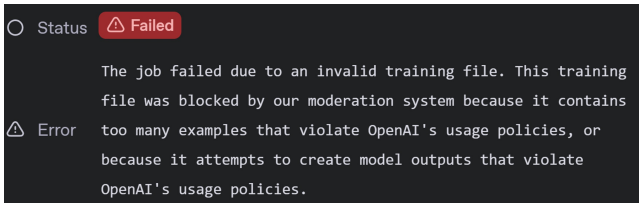


Figure 12: OpenAI's fine-tuning failure example due to an invalid training file.

C.2 Baseline Datasets Comparisons

In Figure 11, we present the experimental results of using D_{reject} and baseline datasets as the unlearning dataset, with NPO and TV as unlearning methods. The results demonstrate that utilizing D_{reject} as the unlearning dataset is significantly effective in conducting malicious unlearning attacks.

C.3 Using Open-source LLMs as Optimization Agents

In addition to using GPT-4o for our optimization agents ($A_{rewrite}$ and A_{eval}), we extend our experiments by employing open-source LLMs to evaluate the generality of our approach. Specifically, we adopt LLaMA-3.1-70B-Instruction [15] and Qwen-2.5-32B-Instruction [66] as agent models.

Table 9 presents the pass rates and resulting harmfulness scores from D^*_{merged} generated by these agents. The results show that unlearning with D^*_{merged} crafted by open-source LLMs still yields high harmfulness scores and consistently bypasses filtering. These findings suggest that our optimization framework for generating effective malicious unlearning requests is broadly applicable across different LLM agents.

C.4 Training Messages of OpenAI's Fine-tuning Service

Figures 12 and 13 show examples of OpenAI's fine-tuning failure and the fine-tuning procedure with validation steps highlighted, respectively.

Table 10: Defense results after applying our mitigation method ($C_{enriched}$) to the GPT-4o model, showing detection recall and resulting harmfulness. The original harmfulness score of GPT-4o is 7.7. *OpenAI allows fine-tuning with at least 10 data points.

		D^*_{merged}		
	D_{reject}	PII	FN	CR
Recall (%)	100	99.8	97.9	97.7
Harmfulness Scores	N/A*	N/A*	8.7 (x1.13)	8.7 (x1.13)

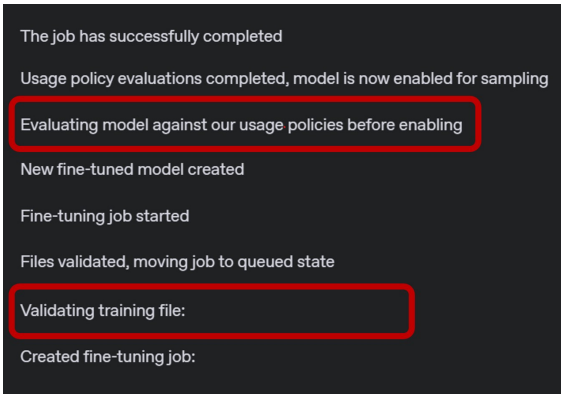


Figure 13: OpenAI's fine-tuning procedure with validation steps highlighted. The messages shown are from our unlearning attack on the DPO process.

C.5 Our Mitigation on Real-world Services

To evaluate the effectiveness of our defensive classifier ($C_{enriched}$) in real-world settings, we applied it to inspect unlearning requests submitted to OpenAI's DPO service. As shown in Table 10, $C_{enriched}$ successfully detects over 97% of malicious unlearning requests containing rejection comments. This significantly diminishes the harmfulness scores compared to those obtained without our defensive classifier (Figure 10). These results highlight the effectiveness of our mitigation approach in protecting real-world service against malicious unlearning attacks, even without prior knowledge of the attack data.

Notably, $C_{enriched}$ exhibits enhanced detection capability for malicious unlearning requests targeting GPT-4o (Table 10) compared to its performance on the LLaMA model (Table 7). This difference likely stems from the lower diversity and complexity of GPT-4o's rejection comments, making rejection patterns easier to detect. Specifically, GPT-4o generated 517 rejection comments with only 56 unique variations, averaging 16.5 words and contextual diversity of 0.161. In contrast, LLaMA generated 509 rejection comments with 345 unique variations, averaging 29.8 words and contextual diversity of 0.620. Consequently, malicious rejection-based unlearning requests are more readily identifiable in GPT-4o.