

Existing Large Language Model Unlearning Evaluations Are Inconclusive

Zhili Feng*
Carnegie Mellon University

Yixuan Even Xu*
Carnegie Mellon University

Alexander Robey
Carnegie Mellon University

Robert Kirk
UK AI Security Institute

Xander Davies
UK AI Security Institute
OATML, University of Oxford

Yarin Gal
UK AI Security Institute
OATML, University of Oxford

Avi Schwarzschild
Carnegie Mellon University

J. Zico Kolter
Carnegie Mellon University

Abstract

Machine unlearning aims to remove sensitive or undesired data from large language models. However, recent studies suggest that unlearning is often shallow, claiming that removed knowledge can easily be recovered. In this work, we critically examine standard unlearning evaluation practices and uncover key limitations that shake our trust in those findings. First, we show that some evaluations introduce substantial new information into the model, potentially masking true unlearning performance by re-teaching the model during testing. Second, we demonstrate that evaluation outcomes vary significantly across tasks, undermining the generalizability of current evaluation routines. Finally, we find that many evaluations rely on spurious correlations, making their results difficult to trust and interpret. Taken together, these issues suggest that current evaluation protocols may both overstate and understate unlearning success. To address this, we propose two principles for future unlearning evaluations: *minimal information injection* and *downstream task awareness*. We validate these principles through a series of targeted experiments, showing how violations of each can lead to misleading conclusions.

1 Introduction

Despite the impressive capabilities of large language models (LLMs), their widespread use introduces significant safety and ethical challenges, particularly regarding their retention of harmful or sensitive knowledge [1]. Concerns regarding the tendency of LLMs to propagate misleading, untrue, or actively harmful content motivate the field of *unlearning* [2]. Unlearning methods, which aim to selectively remove knowledge from LLMs, offer a promising avenue for reducing the amplification of harmful content without the overhead costs associated with retraining models from scratch [3, 4].

Owing to the high-stake nature of deploying unlearning algorithms, it is crucial to design robust evaluation that reliably determine whether the algorithms or resulting models are safe to use. Existing work focuses on evaluations and benchmarks that largely conclude with the claim that unlearning is ineffective [3]. In particular, we find that existing evaluations suffer from two critical shortcomings [e.g. 5–7]. First, they often inject additional information into the model, making it difficult to decouple preexisting knowledge from artifacts of the evaluation process. Second, they frequently employ task-dependent metrics (e.g., multiple-choice question accuracy) that are reductive and fail to account

*Equal contribution. Correspondence to zhilif@andrew.cmu.edu

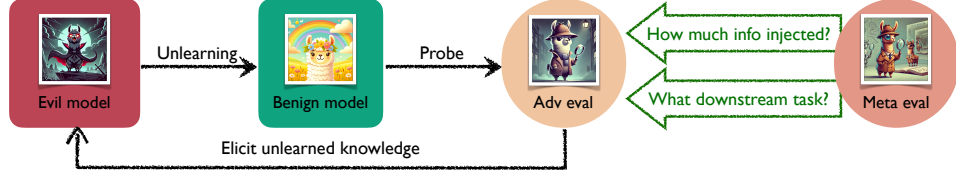


Figure 1: **The evaluation cycle of unlearned models.** Adversarial unlearning evaluations seek to determine whether an unlearned model retains sensitive data after unlearning. In this paper, we show that existing evaluations offer inconclusive results and we propose two principles to improve reliability and ensure future evaluations more accurately reflect true forgetting.

for the diversity of model use cases, and rely on spurious correlations. These limitations undermine the validity of current evaluations, particularly in their capacity to inform model deployment in real-world, safety-critical applications.

Motivated by these shortcomings, we examine two popular existing adversarial unlearning evaluations: input-space attacks and weight-space attacks. The former aims to extract information from the model via input prompts and the latter directly manipulates the model’s parameters, typically through finetuning, to reveal hidden content. We find that prior work proposing a compression-based memorization definition [7] is useful for unlearning evaluation. The analysis of all three unlearning evaluations informs our proposal of two key principles which we argue should guide future evaluations. The *minimal information injection* principle stipulates that evaluations should minimize the amount of information injected via prompts or weight-space manipulations; and the *downstream task awareness* principle states that future evaluations should anticipate a wider scope of model use, including open-ended generation. In proposing these principles, we aim to lay the groundwork for more reliable and actionable assessments of unlearning effectiveness.

2 Related work

The recent literature defines unlearning in different ways [8, 9]. In exact unlearning, the goal is to find a model that is equivalent to a copy of that model trained only on a retain set, which excludes the information to be unlearned [10]. In approximate unlearning, the goal is to remove a training data’s influence statistically, using definition similar to differential privacy [11, 12]. And finally, in heuristic unlearning, the goal is to remove knowledge more abstractly, aiming—like in model alignment [13]—to produce a model that refuses to generate content belonging to a forget set. Each of these paradigms comes equipped with their own evaluation protocols, complicating the task of comparing unlearning algorithms [14, 15].

Exact and approximate unlearning. Several recent pieces of legislation motivate the design of unlearning algorithms. Specifically, an individual’s “right to be forgotten” is covered by a various data protection laws, including the EU’s GDPR (Article 17) [16], the UK GDPR [17], California’s CCPA/CPRA [18], and proposed legislation like Canada’s CPPA [19]—all of which grant individuals the right to request deletion of their personal data. To this end, exact unlearning aims to obtain a model identical to a model trained without a particular piece of data [20–22], whereas approximate unlearning aims to reduce data influence without guaranteeing complete removal [23–26]. In the context of LLMs, learning to “forget” concepts (e.g., information about the Harry Potter book series) has resulted in several finetuning approaches to approximate unlearning [2, 27]. Accompanying these algorithms are various datasets curated to benchmark unlearning in vision and language tasks [28–31]. One notable corpus that we use throughout this work is the Task of Fictitious Unlearning (TOFU) dataset [28], which contains question-answer pairs about distinct, fictitious authors [28].

Heuristic unlearning. A related line of work frames unlearning in the context of model alignment and AI safety [32, 33]. In this spirit, heuristic unlearning algorithms train models via preference optimization to refuse to generate harmful information [4, 34]. Relatedly, there is a notable synergy between representation-based finetuning approaches to improve robustness against jailbreaking [35] and unlearning hazardous knowledge [3]. Tamper-resistant methods, which train models to resist finetuning attacks [5], have also shown promise as unlearning methods in image classification [36] and natural language generation [37, 38]. To measure the performance of heuristic unlearning, numerous

datasets—including variants of the Weapons of Mass Destruction Proxy (WMDP) benchmark [3, 39], which focuses on biological, chemical, cybersecurity risks—are available.

Evaluating unlearning. A central component of unlearning is to conclusively determine whether an unlearned model has forgotten a piece of information. To do so, several papers propose adversarial elicitation methods [5, 6, 39, 40] as well as manual, multi-turn attacks [41]. Che et al. [5] suggest that finetuning attacks tend to upper bound the success rate of adversarial prompting as well as non-adversarial detection methods, including latent-space probing [5]. Relatedly, Schwarzschild et al. [7] propose the adversarial compression ratio as a metric for LLM memorization; their approach provides evidence that in-context unlearning does not effectively unlearn information. That work, and others in the same spirit [42, 43], situate memorization detection as a necessary subtask in unlearning evaluations, which generally involve demonstrating *both* that the forget set is no longer memorized and that the retain set remains learned.

Several recent studies examine the fragility of LLM unlearning evaluations. Hu et al. [44] find that an unlearned model can be retrained on a small, unrelated dataset to output harmful knowledge that it had supposedly forgotten. Thaker et al. [14] also show that unlearning algorithms tend to overfit to narrowly defined retain and forget sets, and that non-adversarial query changes tend to significantly change unlearning success metrics. In contrast, in our work, we focus our analysis on stronger finetuning and adversarial attacks, both of which are pervasive in the unlearning literature [2].

3 Preliminaries

Terminology. We use the following standard terminology to describe unlearning datasets and models. We refer to the dataset intended for removal as the *forget set*, and to the remainder of the training data, which the model should retain, as the *retain set*. The model before unlearning is the *base model*, the result of unlearning is the *unlearned model*, and a model trained from scratch on only the retain set is the *retain model*. In finetuning-based attacks, we call the model obtained by finetuning the unlearned model the *relearned model*, and the corresponding procedure is called *relearning*.

3.1 Unlearning evaluation methods

Drawing from the taxonomy laid out by Che et al. [5], we center our analysis on prominent methods from three broad, representative classes of unlearning evaluations: finetuning attacks, input-space attacks, and memorization detectors. In the following subsections, we describe each evaluation, justify its choice, and provide in-depth preliminaries and notation specific to each attack.

Finetuning attacks. Finetuning attacks allow an adversary to finetune the unlearned model on several specifically chosen samples, a process we refer to as relearning. In practice, these training samples can be drawn from either the retain set or the forget set; in this work, we focus on samples drawn from the retain set, as they give stronger evidence of unlearning success or failure. To support this choice, consider the work of Łucki et al. [6], who find that an unlearned model trained via RMU [3] to forget WMDP-Bio achieves only 29.9% accuracy on questions from this forget set. However, after relearning on just *five* samples from the retain set, the accuracy on the forget set spikes to 62.4%, whereas the base model only achieves 64.4% accuracy before unlearning anyway. This indicates that finetuning on a remarkably small subset of the retain set is sufficient to recover nearly all the accuracy of the base model, which evinces relatively weak unlearning in this case. We note that finetuning attacks establish an upper bound on the success rate of other adversarial evaluations, as they offer the greatest flexibility by directly modifying the model’s weights [5].

Input-space attacks. Whereas finetuning attacks edit model internals to probe relearning capabilities, input-space attacks seek to elicit supposedly unlearned text via prompting [45–47]. In this paper, we consider the Enhanced GCG attack, which is representative, state-of-the-art algorithm in this category [6]. The objective of Enhanced GCG is to optimize a single adversarial prompt, which, when prepended to any prompt in the forget set, facilitates the elicitation of unlearned knowledge. To operationalize this attack, let M and M_U denote a base and unlearned model, respectively, and let D denote a given dataset, which can be chosen somewhat arbitrarily, but is often taken to be a subset of the retain set or the forget set. Enhanced GCG then seeks to solve the following problem:

$$\underset{x}{\text{maximize}} \quad \frac{1}{|D|} \sum_{y \in D} \log \Pr(x||y; M_U) + \lambda(x; D, M, M_U). \quad (1)$$

Here, $\cdot||\cdot$ denotes the concatenation operator and x is the optimization variable. In [6], they also constraint the length of x to be 100. $\Pr(x||y; M_U)$ denotes the probability of the unlearned model generating a piece of knowledge that depends on y , and $\lambda(x; D, M, M_U)$ term is a regularization term that encourages the prefix x be such that the internal representations of M_U are relatively close to those of M [48]. Given a solution x^* to (1), one can evaluate M_U on a held-out set.

Memorization detectors. Memorization detection is a fundamental piece of any unlearning evaluation pipeline. While various metrics exist for quantifying memorization, one prominent metric is the adversarial compression ratio (ACR) [7]. And although they do not typically frame ACR as an unlearning metric, Schwarzschild et al. [7] have a finding where it is used to evince failures to forget information across various unlearned models. For example, they highlight that the ACR of Harry Potter information that Eldan and Russinovich [27] try to remove from an LLM remains unchanged. Given a model M (either a base or unlearned model), the ACR of a string y is defined as

$$\text{ACR}(M, y) = |y|/|x^*|, \quad \text{where } x^* \in \arg \min |x| \text{ s.t. } M(x) = y. \quad (2)$$

Here $|\cdot|$ denotes the length of a sequence of tokens, and $M(x) = y$ indicates that M generates y in response to a prompt x under greedy decoding. To implement a complete evaluation of unlearning via the ACR, we compare the ACR measured separately for both the retain and forget sets to measure unlearning success on a per-sample basis, as opposed to Enhanced GCG.

3.2 Datasets, architectures, and unlearning algorithms

We use a variety of standard unlearning algorithms, datasets, and LLM architectures to facilitate our analysis of the effectiveness of current unlearning evaluations. Following Łucki et al. [6], we use Zephyr-7B [49] as the base model in many of our experiments, and to offer points of comparison, we also evaluate Phi-1.5 [50] and Llama-3.2-1B-Instruct models [51] when applicable. All chat models are evaluated with empty system prompts. To obtain unlearned models, we use two standard algorithms: representation misdirection for unlearning (RMU) [3] and negative preference optimization (NPO) [4]. When evaluating unlearning, we use the standard splits of both TOFU [28] and WMDP [3], unless otherwise stated. Our evaluations of finetuning attacks also use a newly curated multiple-choice question (MCQ) version of TOFU, which we call TOFU-MCQ. This data comprises 2000 generated MCQs—ten MCQs for each of TOFU’s 200 fictitious authors. We evaluate these behaviors on Phi-1.5, which, crucially, was released *before* the curation of the TOFU.

4 Pitfalls of adversarial unlearning evaluations

We begin by identifying and analyzing the shortcomings of the first two adversarial evaluation methods discussed in §3.1: finetuning attacks and input-space attacks. Our analysis centers on two key questions, which motivate the recommendations we make in later sections below.

Question 1: Does the information elicited by an adversarial evaluation reflect a failure to unlearn the forget set or new information internalization introduced by the attack process itself?

Question 2: To what extent is the effectiveness of adversarial evaluations sensitive to the formatting of the task (e.g., MCQs versus open-ended generation)?

The first question posits two possible sources for the information elicited in an unlearning evaluation: text that the unlearned model failed to forget, and text introduced inadvertently during evaluation. While identification of the first source conclusively points to ineffective unlearning, distinguishing it from the second is essential for correctly attributing failure and avoiding false positives in evaluation. The second question, on the other hand, concerns the sensitivity of standard evaluations. If evaluating unlearning via open-ended generation yields significantly different conclusions than an equivalent evaluation performed with MCQs, it suggests that conclusions about unlearning success may be tightly coupled to the chosen task format, limiting their reliability. We consider these questions for finetuning attacks in §4.1 and for input-space attacks in §4.2.

4.1 Finetuning attacks

Finetuning-based unlearning evaluations seek to adjust an unlearned model by training on a minimal number of samples. The resulting relearned model is then evaluated to determine whether it contains

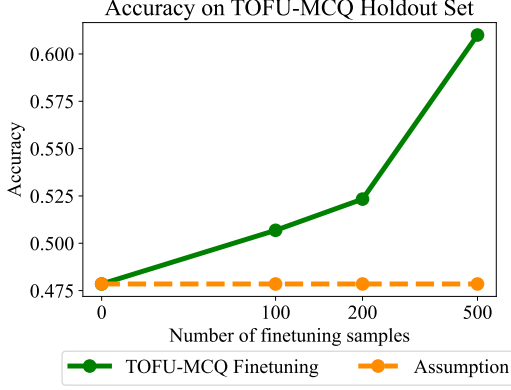


Figure 2: **Evidence of spurious generalization.** Finetuning attacks significantly improve accuracy on TOFU’s test set. This finding indicates that it may be possible to spuriously generalize between the retain and forget sets in finetuning-based unlearning evaluations.

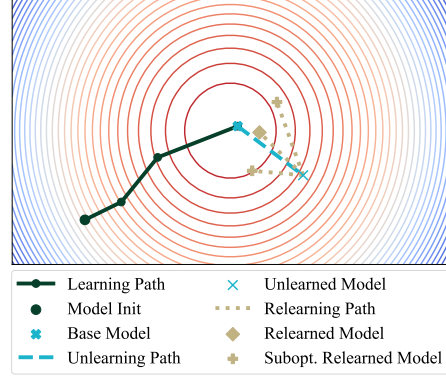


Figure 3: **Relearning trajectories.** An illustrative case in which two phenomena occur: (1) a relearned model has similar performance to the base model, and (2) efficiently finding the relearning path relies on prior knowledge of the unlearning algorithm and forget data.

knowledge about the forget set. The fidelity of this evaluation rests on two key assumptions: (1) the relearned model should not generalize to the forget data purely from training on the retain data, and (2) the evaluator knows the data format used by the unlearning algorithm. In the remainder of this subsection, we empirically show that neither of these assumptions may hold in practice.

Evidence of spurious generalization. Unlearning evaluations implicitly assume that the retain and forget sets are disjoint and independent. If this property does not hold, successful unlearning may be impossible, since unlearning the forget set may diminish a model’s knowledge of the retain set, and conversely, fine-tuning on the retain set may result in relearning the forget set. To test whether this form of generalization manifests in existing finetuning evaluations, we design the following experiment. We first finetune Phi-1.5 on subsets of TOFU-MCQ containing 100, 200, and 500 instances. Next, we evaluate these finetuned models on held-out data from TOFU-MCQ. Since each instance in TOFU-MCQ comprises information about distinct people, one would expect generalization between the finetuning and held-out sets to be impossible. However, as shown in Figure 2, we find that finetuning significantly improves accuracy on the held-out set. This implies that TOFU, a widely used unlearning benchmark, contains spurious correlations that facilitate generalization between the retain and forget sets. In other words, finetuning a model that has never seen the forget set on retain data only may introduce knowledge about the forget set into the model, complicating the task of identifying successful unlearning.

Evidence of data formatting dependence. A critical, yet often overlooked aspect of unlearning is the role played by prior knowledge. Both unlearning and evaluation require a (somewhat arbitrary) choice of the unlearning algorithm (e.g., RMU [3] or DPO [4]), the data format (e.g., MCQ or full corpora), and the evaluation metric. In the case of finetuning-based evaluations (Figure 3), we illustrate how the relearning path may depend on the unlearning algorithm and data format. In particular, this example shows that numerous relearned finetuning trajectories are possible, and that prior knowledge of the forget set may influence the extent to which a relearned model approaches a checkpoint similar to the base model. Furthermore, we note that numerous studies on finetuning attacks implicitly assume that the data formats match between training and evaluation (see, e.g., [5, 6, 28, 29]).

To probe the relationship between evaluation results and data format, we consider the following simple experiment. We first take two unlearned models: one is unlearned via NPO on WMDP-Bio MCQ data and the other is unlearned via RMU on WMDP-Bio full corpora data. Next, we finetune both models separately on the retain set of WMDP-Bio MCQ and WMDP-Bio corpora data, which yields four distinct relearned checkpoints. The results of this experiment, which are shown in Figure 4, show that relearning effectiveness varies significantly when the finetuning data differs in format or structure from the original unlearning data, even if they encode the same information. In particular, we observe that relearning tends to require fewer samples when the unlearning and relearning data

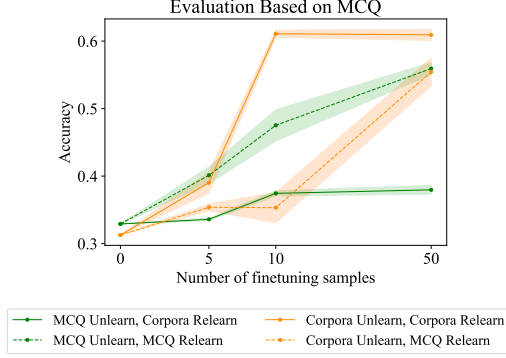


Figure 4: **Evidence of information injection.** Re-learning effectiveness varies significantly when the data format of the downstream task generation format differs between evaluation and deployment.

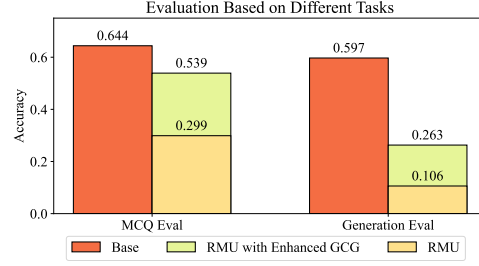


Figure 5: **Enhanced GCG task dependence.** The effectiveness of Enhanced GCG varies widely across downstream tasks. Whereas this method nearly recovers the base model’s accuracy for maximum letter probability generation, it offers little improvement in accuracy for maximum text generation tasks.

formats match. This implies that successful evaluations may require prior knowledge about the original unlearning algorithm.

4.2 Input-space attacks

While it may be intuitive that finetuning evaluations inject new knowledge into an unlearned model, what may be less clear is that prompting presents another opportunity for the inadvertent injection of new information. In this subsection, we offer evidence that Enhanced GCG, a standard optimization-based input-space attack, tends to encode substantial information about the forget set, rendering the results of these evaluations inconclusive.

Evidence of information injection. As described in §3.1, Enhanced GCG optimizes a string of tokens that, when prepended to an input prompt, facilitates the elicitation of knowledge about the forget set. However, as LLMs are universal sequence approximators [52], one cannot immediately rule out the possibility of a well-optimized prefix injecting new information during evaluation. To illustrate this point, consider that in their evaluations, Lucki et al. [6] optimize Enhanced GCG strings comprising 100 tokens on fewer than ten samples. They ultimately find that this optimization pressure is sufficient to achieve nearly 55% accuracy on the WMDP-Bio MCQ test set. This split of WMDP-Bio contains fewer than 1300 samples, and therefore encoding each of the four possible answers to each MCQ question requires approximately $1300 \times \log 4 \times 0.55 \approx 1430$ bits. On the other hand, given that the vocabulary size of the model they use is nearly 32,000, the 100-token adversarial strings encode roughly $100 \times \log 32000 \approx 1500$ bits of information. This estimate suggests that the adversarial prompts may already embed sufficient information to elicit the correct answers, undermining the validity of the evaluation. It is thus unclear whether the prompt is reactivating superficially unlearned knowledge or introducing new knowledge into the model.

To support this hypothesis, we show that one can optimize prompts to boost performance on the forget set. To this end, we first finetune Llama-3.2-1B-Instruct to the point of overfitting on the WMDP-Bio test set. We then optimize an Enhanced GCG prefix on this finetuned model using seven samples from WMDP-Bio. Whereas the base model achieves 39.9% accuracy on the entire WMDP-Bio, we find that when the optimized prefix is prepended to the same queries, accuracy of the same base model reaches 53.8%. This spike in performance suggests that input prompt optimization can indeed inject knowledge about the forget set directly into the model.

4.3 A common confounder: MCQ formatting

Numerous unlearning evaluations use data formatted as MCQs [3]. And while this format is compact and intuitive, we find that *both* finetuning and input prompt evaluations can lead to inconclusive results depending on how the answers to each MCQ are calculated. To demonstrate this, assume that

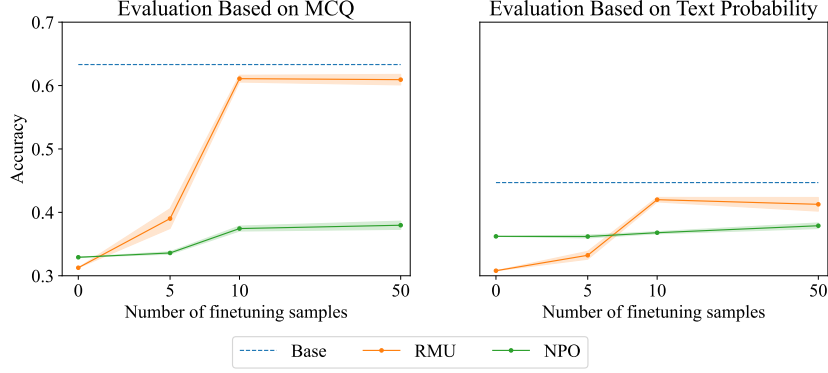


Figure 6: **Finetuning attacks across MCQ formats.** While finetuning attacks computed for maximum letter probability generation recover the base model’s performance, analogous attacks on the maximum text probability do not yield similar accuracy recovery.

we are given a model M , an MCQ string q , and four answer strings a_1, a_2, a_3 , and a_4 . Define

$$p = \{q\} \backslash \text{nA} . \{a_1\} \backslash \text{nB} . \{a_2\} \backslash \text{nC} . \{a_3\} \backslash \text{nD} . \{a_4\} \backslash \text{nAnswer} : \quad (3)$$

For example, an MCQ from WMDP-Bio formatted as in the same way as p is as follows:

The following are multiple choice questions (with answers) about biology. Which animal model was used in the study to assess chronic renal insufficiency? \nA. Pigs \nB. Goats \nC. Rats \nD. Mice \nAnswer:

There are (at least) two reasonable ways to calculate the model’s answer to q . The first way involves the calculation of what we term the *maximum letter probability*, which is computed by solving

$$\arg \max_{\ell} \Pr(\ell|p; M) \quad \text{subject to} \quad a \in \{A, B, C, D\}, \quad (4)$$

where $\Pr(\ell|p; M)$ denotes the probability of the letter ℓ being the next token predicted by M in response to the input prompt p . In this way, (4) represents the model’s greedy prediction of the letter corresponding to the correct answer. A second reasonable way to compute the answer is to calculate the *maximum text probability*, which is computed by solving

$$\arg \max_a 1/|a| \cdot \log \Pr(a|q; M) \quad \text{subject to} \quad a \in \{a_1, a_2, a_3, a_4\}. \quad (5)$$

Thus, (5) represents the average log probability of the predicted answer’s text, rather than the corresponding letter.

Inconclusive finetuning attacks. In the context of finetuning attacks, the results in Figure 6 (left panel) show that unlearned models evaluated by computed the maximum letter probability tend to recover the base model’s MCQ accuracy on WMDP-Bio. However, when answers to the same set of questions are computed via the maximum text probability, Figure 6 (right panel) shows an analogous trend does *not* hold, particularly for NPO models. This implies that MCQ evaluation accuracy is highly dependent on the method used to calculate the answers.

Inconclusive input-space attacks. The results in Figure 5 show an analogous trend for input prompt attacks. When answers are calculated via the maximum letter probability, we find that Enhanced GCG evaluations record 53.9% accuracy on the WMDP-Bio test set. However, when the answers are computed by validating the open-ended generation of the model, the accuracy after applying the Enhanced GCG prefix only results in 26.3% accuracy on the same dataset. This suggests that Enhanced GCG yields prefix strings, or input-space attacks, that are sensitive to the output format. Thus, we cannot conclude that some information is really present in the model, as these Enhanced GCG attacks may be overstating knowledge content by boosting MCQ accuracy only.

Inconclusive memorization detectors. We next design a set of analogous experiments for the ACR memorization detector described in §3.1. Specifically, we consider three Zephyr-7B checkpoints: the base model and two unlearned models obtained by running RMU and NPO. We evaluated these models on the first 100 questions from WMDP-Bio, WMDP-Chem, and WMDP-Cyber. Answers

to the WMDP MCQ questions were generated in three ways: (1) the maximum letter probability as described in (4) (termed “CHOOSE”), (2) a variant of the maximum letter probability where we determine whether the correct answer has the highest log-probability over the model’s entire vocabulary (termed “OPTION”), and (3) determining whether the response generated via greedy decoding corresponds to the text of the correct answer (termed “GENERATE”). For each combination of model, dataset, and downstream task, we use the ACR metric to compute a minimal length suffix for each sample that maximizes the probability of generating the correct answer. The evaluation is considered successful if the length of the suffix is less than a fixed task-specific threshold.

The results of this experiment, which are shown in Figure 7, indicate that conclusions regarding the relative effectiveness of unlearning algorithms can be drastically different across downstream tasks. For the CHOOSE task, both methods offer a relatively small, though non-negligible reduction in the success rate. In contrast, for the OPTION task, both RMU and NPO significantly reduce the success rates. And finally, the results for the GENERATE task indicate that RMU decreases the success rate, whereas NPO increase the success rate relative to the base model. Collectively, these results show that conclusions regarding unlearning vary widely depending on the downstream task.

5 Principles for conclusive unlearning evaluations

The evidence presented in §4 indicates that existing unlearning evaluations are often inconclusive. To support the development of more effective evaluations in future research, we propose two guiding principles—*minimal information injection* and *downstream task awareness*—which we summarize below and describe in more detail in the ensuing section.

Principles for conclusive unlearning evaluations

1. **Minimal information injection:** Unlearning evaluations should not enable the injection of additional information into an unlearned model.
2. **Downstream task awareness:** Unlearning evaluations should be designed to accommodate the ways in which future users will interact with an unlearned model.

Minimal information injection. In §4, we observed that both finetuning attacks and input-prompt attacks directly inject new information into an unlearned model. Thus, we argue that evaluations should minimize the amount of information unintentionally injected into the model. This principle—which we term *minimal information injection*—ensures that the generations of an unlearned model are a realistic reflection of the model’s knowledge, rather than an artifact of the evaluation process.

Downstream task awareness. In §4, we also found that standard unlearning evaluations yield varying results depending on the task in which the unlearned model may ultimately be deployed. More specifically, we found that unlearning results varied depending on whether a model was evaluated via open-ended generation, maximum letter probability, or maximum text probability. We therefore advocate that unlearning evaluations be designed to accommodate the various ways in which future users will interact with an unlearned model, a property we term *downstream task awareness*.

5.1 Recommendations for future evaluations

Recommendation 1: Disclosure of an “injection budget.” Throughout this work, we measure information injection by counting the number of bits available to an attacker and by measuring spurious generalization during relearning. Both of these metrics are relatively heuristic, which is indicative of the inherent difficulty in accurately quantifying how efficiently an LLM processes and stores information [53]. We therefore argue that future work should (a) seek to measure of injected information, such as the heuristic measures or other tools in the information theoretic literature surrounding LLMs [54–56], and (b) allow adversaries a fixed (metric-dependent) budget for information injection.

Recommendation 2: Report cross-modality leakage metrics. Task sensitivity in unlearning evaluations is indicative of a wider trend; recent findings have noted similar robustness concerns in tasks beyond unlearning [57–59]. And indeed, task sensitivity is not necessarily a pitfall in tasks where a particular data format is generally preferred. However, the central goal of unlearning is to ensure that the model cannot generate any information from the forget set, regardless of its

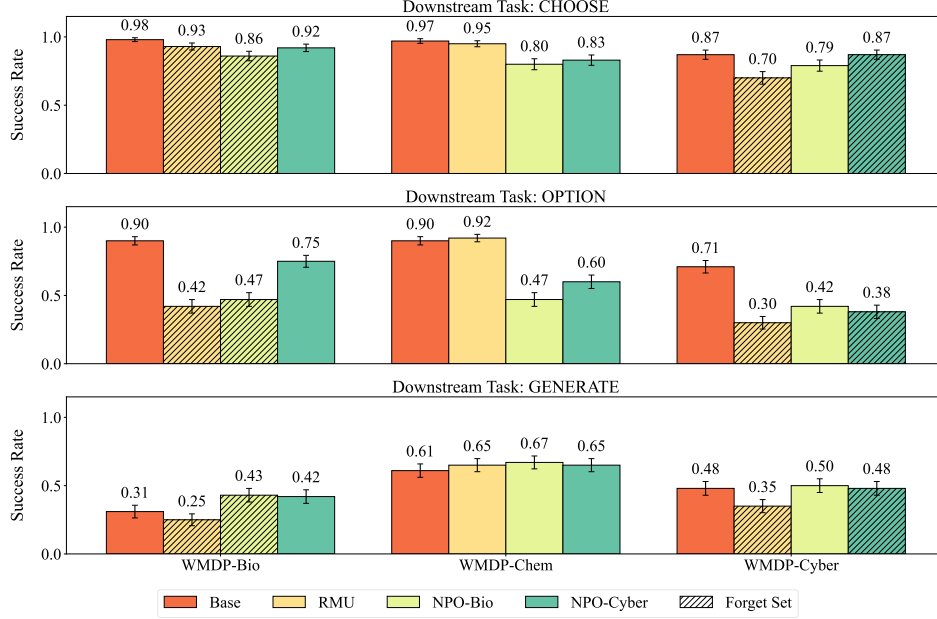


Figure 7: **Inconclusive ACR evaluations on WMDP-Bio.** Success rates of different models on different datasets and tasks. The error bars show the standard errors. For each task, the conclusion about the relative effectiveness of unlearning methods differ. **Top:** CHOOSE task. The success rates of the base and unlearned models are relatively similar; unlearning tends to decrease the success rate marginally. **Middle:** OPTION task. The success rates of the unlearned models are significantly lower than the base model. **Bottom:** GENERATE task. The RMU model has a lower success rate than the base model on WMDP-Bio and WMDP-Cyber. However, NPO tends to increase success rates.

format. Therefore, in line with the downstream task awareness principle, we recommend that future evaluations report a cross-format leakage matrix, which, as in Figure 7, probes an unlearned model for unlearned information across data formats. This standard contributes to a research environment in which describing a model as “unlearned” is more compelling, because a model that can still generate forgotten information in a different format has not truly unlearned.

Recommendation 3: Use memorization detectors as a yard-stick. Our focus is on detecting the presence or absence of information, which is related to but distinct from memorization. While the distinction is subtle, memorization tools—such as the ACR—can still aid unlearning evaluation. In particular, determining whether a sample is memorized relates to a stricter criterion, so a positive result strongly suggests the information is present, even if a negative result remains inconclusive. Our recommendation is incorporate tests from the memorization space as high bars to clear when it comes to unlearning—if a training sample is memorized by some definition after unlearning, we should conclude that unlearning failed.

6 Conclusion

In this work, we identify the shortcomings in existing adversarial evaluations for safety unlearning and propose two guiding principles for more reliable assessment methods. We encourage future research to develop more efficient and principled evaluation techniques.

Broader impact. The AI safety community is contending with a rapidly developing landscape of products and services and a seriously understudied risk landscape. As regulation and policy emerges to protect users and the field proposes techniques to implement mandated features, it is critical that we carefully evaluate new methods that purport to preserve or protect privacy. Our work sits at this boundary, aiming to shore up the way we audit, monitor, and evaluate critical aspects of the developing AI infrastructure. With this in mind, the impact of our work extends beyond the academic sphere and the limitations must therefore be stated and understood clearly. We carry out

experiments with a small number of unlearning methods and only academic style benchmark datasets. Our findings serve as a demonstration that existing evaluations have flaws, and this proof of existence is sound even with our limited experimental scope. Our hope is that the recommendations we make and the evaluation principles we advocate are impactful at the interface of AI and society and that our work is useful to practitioners and policy makers, alike.

Acknowledgments

ZF, YEX, and AS are supported by the Bosch Center for Artificial Intelligence. YEX also acknowledges support from the NSF through grant IIS-2200410. AR is supported by ONR award N000142412693. Finally, ZK gratefully acknowledges the Bosch Center for Artificial Intelligence for its support of the work in his lab as a whole.

References

- [1] Nicholas Carlini, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei W Koh, Daphne Ippolito, Florian Tramèr, and Ludwig Schmidt. Are aligned neural networks adversarially aligned? *Advances in Neural Information Processing Systems*, 36, 2024.
- [2] 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. *Nature Machine Intelligence*, pages 1–14, 2025.
- [3] 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.
- [4] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.
- [5] Zora Che, Stephen Casper, Anirudh Satheesh, Rohit Gandikota, Domenic Rosati, Stewart Slocum, Lev E McKinney, Zichu Wu, Zikui Cai, Bilal Chughtai, et al. Model manipulation attacks enable more rigorous evaluations of llm capabilities. In *Neurips Safe Generative AI Workshop*, 2024.
- [6] Jakub Łucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. An adversarial perspective on machine unlearning for ai safety. *arXiv preprint arXiv:2409.18025*, 2024.
- [7] Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary C Lipton, and J Zico Kolter. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint arXiv:2404.15146*, 2024.
- [8] Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.
- [9] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX security symposium (USENIX Security 22)*, pages 4007–4022, 2022.
- [10] Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. Arcane: An efficient architecture for exact machine unlearning. In *IJCAI*, volume 6, page 19, 2022.
- [11] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.
- [12] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11516–11524, 2021.
- [13] 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.

- [14] Pratiksha Thaker, Shengyuan Hu, Neil Kale, Yash Maurya, Zhiwei Steven Wu, and Virginia Smith. Position: Llm unlearning benchmarks are weak measures of progress. *arXiv preprint arXiv:2410.02879*, 2024.
- [15] Yan Scholten, Stephan Günnemann, and Leo Schwinn. A probabilistic perspective on unlearning and alignment for large language models. *arXiv preprint arXiv:2410.03523*, 2024.
- [16] European Union. Regulation (eu) 2016/679 of the european parliament and of the council. *Official Journal of the European Union*, 2016.
- [17] UK Government. UK General Data Protection Regulation (UK GDPR), 2021. URL <https://www.gov.uk/data-protection>. Post-Brexit adaptation of the EU GDPR, enforced by the UK Information Commissioner’s Office.
- [18] CA OAG. Ccpa regulations: Final regulation text. *Office of the Attorney General, California Department of Justice*, 2021.
- [19] Government of Canada. Bill C-27: Digital Charter Implementation Act, 2022 – Consumer Privacy Protection Act (CPPA), 2022. URL <https://www.parl.ca/DocumentViewer/en/44-1/bill/C-27/first-reading>. Proposed legislation to modernize Canadian privacy law, including rights to erasure and AI oversight.
- [20] Enayat Ullah, Tung Mai, Anup Rao, Ryan A Rossi, and Raman Arora. Machine unlearning via algorithmic stability. In *Conference on Learning Theory*, pages 4126–4142. PMLR, 2021.
- [21] Lucas Bourtole, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.
- [22] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 383–398. Springer, 2020.
- [23] 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.
- [24] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.
- [25] 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.
- [26] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.
- [27] Ronen Eldan and Mark Russinovich. Who’s harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.
- [28] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.
- [29] Yingzi Ma, Jiong Xiao Wang, Fei Wang, Siyuan Ma, Jiazhaoli, Xiujun Li, Furong Huang, Lichao Sun, Bo Li, Yejin Choi, Muhao Chen, and Chaowei Xiao. Benchmarking vision language model unlearning via fictitious facial identity dataset, 2024. URL <https://arxiv.org/abs/2411.03554>.
- [30] 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.

- [31] 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.
- [32] Fazl Barez, Tingchen Fu, Ameya Prabhu, Stephen Casper, Amartya Sanyal, Adel Bibi, Aidan O’Gara, Robert Kirk, Ben Bucknall, Tim Fist, et al. Open problems in machine unlearning for ai safety. *arXiv preprint arXiv:2501.04952*, 2025.
- [33] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwal, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramèr, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024.
- [34] Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *arXiv preprint arXiv:2410.07163*, 2024.
- [35] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [36] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [37] Peter Henderson, Eric Mitchell, Christopher Manning, Dan Jurafsky, and Chelsea Finn. Self-destructing models: Increasing the costs of harmful dual uses of foundation models. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 287–296, 2023.
- [38] Rishub Tamirisa, Bhargu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, et al. Tamper-resistant safeguards for open-weight llms. *arXiv preprint arXiv:2408.00761*, 2024.
- [39] Aggyad Deeb and Fabien Roger. Do unlearning methods remove information from language model weights? *arXiv preprint arXiv:2410.08827*, 2024.
- [40] Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.
- [41] Nathaniel Li, Ziwen Han, Ian Steneker, Willow Primack, Riley Goodside, Hugh Zhang, Zifan Wang, Cristina Menghini, and Summer Yue. Llm defenses are not robust to multi-turn human jailbreaks yet. *arXiv preprint arXiv:2408.15221*, 2024.
- [42] Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.
- [43] Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53. Association for Computational Linguistics, 2023.
- [44] Shengyuan Hu, Yiwei Fu, Zhiwei Steven Wu, and Virginia Smith. Unlearning or Obfuscating? Jogging the Memory of Unlearned LLMs via Benign Relearning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL <https://arxiv.org/abs/2406.13356>. *arXiv:2406.13356*.
- [45] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

- [46] Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.
- [47] Alexander Robey, Zachary Ravichandran, Vijay Kumar, Hamed Hassani, and George J Pappas. Jailbreaking llm-controlled robots. *arXiv preprint arXiv:2410.13691*, 2024.
- [48] T Ben Thompson and Michael Sklar. Flrt: Fluent student-teacher redteaming. *arXiv preprint arXiv:2407.17447*, 2024.
- [49] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Cl  mentine Fourier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023.
- [50] Yuanzhi Li, S  bastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report, 2023. URL <https://arxiv.org/abs/2309.05463>.
- [51] 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.
- [52] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? *arXiv preprint arXiv:1912.10077*, 2019.
- [53] Zorik Gekhman, Eyal Ben David, Hadas Orgad, Eran Ofek, Yonatan Belinkov, Idan Szpektor, Jonathan Herzig, and Roi Reichart. Inside-out: Hidden factual knowledge in llms. *arXiv preprint arXiv:2503.15299*, 2025.
- [54] Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. *arXiv preprint arXiv:2004.03061*, 2020.
- [55] Hang Chen, Xinyu Yang, Jiaying Zhu, and Wenya Wang. Quantifying emergence in large language models. *arXiv preprint arXiv:2405.12617*, 2024.
- [56] Pouya Pezeshkpour. Measuring and modifying factual knowledge in large language models. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 831–838. IEEE, 2023.
- [57] Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*, 2023.
- [58] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. *arXiv preprint arXiv:2309.03882*, 2023.
- [59] Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. Does prompt formatting have any impact on llm performance? *arXiv preprint arXiv:2411.10541*, 2024.

A Additional Experiments

In this section, we present additional results in the set of experiments described in Section 5. To interpret the results, we note that the unlearned models we use in Section 5 are only unlearned on a subset of the WMDP dataset. Specifically, the RMU model is unlearned on both WMDP-Bio and WMDP-Cyber, while the NPO model is unlearned on WMDP-Bio. In the full experiments, we include another model that is unlearned on WMDP-Cyber using the NPO method. We differentiate the NPO models unlearned on WMDP-Bio and WMDP-Cyber by referring to them as NPO-Bio and NPO-Cyber, respectively.

A.1 Success Rates for Different Downstream Tasks

In Figure 7, we show the success rates on WMDP for 3 different downstream tasks mentioned in Section 4.3. The results confirm that conclusions about the effectiveness of unlearning methods can be drastically different for different downstream tasks. For the CHOOSE task, both RMU and NPO reduce the success rate on the datasets they unlearn on. However, neither of them decrease it significantly. For the OPTION task, both RMU and NPO reduce the success rate significantly, with RMU reducing it slightly more. For the GENERATE task, RMU reduces the success rate, while NPO increases it.

On a side note, we also remark that none of the models are unlearned on WMDP-Chem, so we could consider WMDP-Chem as a retain set. As we can see, for the CHOOSE and OPTION tasks, the success rate of the RMU model on WMDP-Chem is similar to the base model, while the NPO models have lower success rates on WMDP-Chem. This suggests that RMU preserves the model’s performance on benign knowledge better than NPO.

A.2 Adversarial Compression Ratio for Different Downstream Tasks

In Table 1, we show the 40%, 50% and 60% percentiles of adversarial compression ratio (ACR) of the models on WMDP for the GENERATE task. The results show that RMU has a better effect of unlearning. The RMU model has a lower ACR than the Base model on WMDP-Bio and WMDP-Cyber. The NPO models also reduce ACRs, but the magnitude is smaller. This indicates that it is harder to extract hazardous knowledge from the RMU model, which concurs with the results in Figure 7. Moreover, looking at the results of WMDP-Chem—data that is not included in any forget sets in this experiment—we can also see that the RMU model and the Base model’s performance are consistently relatively similar on knowledge that is not targeted in the unlearning process, while the NPO models tend to decline in performance on such knowledge. This suggests that RMU more effectively preserves the model’s performance on benign knowledge. Overall, when considering open-ended generation as the downstream task, RMU does seem to have a better effect of unlearning.

Table 1: 40%, 50% and 60% percentiles of adversarial compression ratio (ACR) of 4 variants of Zephyr 7B on WMDP for the GENERATE task. Performance is compared against Base. Green cells highlight the models unlearned on the corresponding datasets. Higher intensity of the color means it is harder to extract the knowledge. The results show that RMU has a better effect of unlearning.

Dataset	Base	RMU	NPO-Bio	NPO-Cyber
WMDP-Bio	2.00 / 2.50 / 2.87	0.85 / 1.25 / 1.87	1.50 / 1.58 / 1.87	1.50 / 2.00 / 2.00
WMDP-Chem	1.50 / 1.67 / 2.00	1.50 / 1.67 / 1.77	1.50 / 1.57 / 1.67	1.43 / 1.63 / 2.00
WMDP-Cyber	2.38 / 3.00 / 3.00	1.56 / 1.83 / 3.00	2.00 / 2.10 / 2.70	2.20 / 2.75 / 3.00

B Experiment Details

For the experiments involving finetuning on WMDP data in Section 3.1, we always use LoRA rank 128, $\alpha = 16$, learning rate $2e-4$, and we always train for 3 epochs with batch size 1. For the experiments with ACR in Section 3.1, we use learning rate $1e-2$, batch size 100, and the 250 top choices for the optimization step. For the CHOOSE and OPTION tasks, we run 200 steps of optimization with 5 free tokens. For the GENERATE task, we run 350 steps with 20 free tokens. We choose these thresholds such that further increasing them does not boost the success probability of

the corresponding task significantly further. For the TOFU experiment, we train with LoRA rank 128, $\alpha = 16$, learning rate $2e-4$, and we always train for 5 epochs with batch size 16.

For all our experiments, we use one A6000 with 48GB of memory. The experiments in Section 4 take around 20 GPU hours, and the experiments in Section 5 take approximately 800 GPU hours.