Does Machine Unlearning Truly Remove Knowledge?

Haokun Chen[1,2,4] **Yueqi Zhang**[3] **Yuan Bi**[3] **Yao Zhang**[1,2] **Tong Liu**[1,2] **Jinhe Bi**[1,2] **Jian Lan**[1,2]
Claudia Grosser[3,4] **Denis Krompaß**[4] **Jindong Gu**[5] **Nassir Navab**[3] **Volker Tresp**[1,2]
[1]LMU Munich [2]Munich Center for Machine Learning (MCML)
[3]Technical University of Munich [4]Siemens AG [5]University of Oxford

Abstract

In recent years, Large Language Models (LLMs) have achieved remarkable advancements, drawing significant attention from the research community. Their capabilities are largely attributed to large-scale architectures, which require extensive training on massive datasets. However, such datasets often contain sensitive or copyrighted content sourced from the public internet, raising concerns about data privacy and ownership. Regulatory frameworks, such as the General Data Protection Regulation (GDPR), grant individuals the right to request the removal of such sensitive information. This has motivated the development of machine unlearning algorithms that aim to remove specific knowledge from models without the need for costly retraining. Despite these advancements, evaluating the efficacy of unlearning algorithms remains a challenge due to the inherent complexity and generative nature of LLMs. In this work, we introduce a comprehensive auditing framework for unlearning evaluation, comprising 3 benchmark datasets, 6 unlearning algorithms, and 5 prompt-based auditing methods. By using various auditing algorithms, we evaluate the effectiveness and robustness of different unlearning strategies. To explore alternatives beyond prompt-based auditing, we propose a novel auditing technique based on intermediate activation perturbation. This approach offers a new perspective and serves as a potential direction for the future design of auditing algorithms. The complete framework and the proposed algorithm will be open-sourced upon manuscript acceptance.

1 Introduction

Large language models (LLMs) have seen rapid advancements recently, resulting in improved performance and widespread adoption across numerous applications. These advancements are largely attributed to their large-scale architectures, which require training on datasets containing billions of tokens [15]. These datasets are typically constructed from large-scale corpora of publicly available internet text. However, such corpora often inadvertently include personally identifiable information (PII) or copyrighted material, which are considered sensitive and generally unsuitable for commercial use due to legal and ethical constraints. To comply with local regulations (e.g., GDPR [16]) and internal policies, it is often necessary to remove sensitive information from trained models.

Machine unlearning has emerged as a promising solution to this problem [5, 4]. This work is motivated by the legal framework proposed by the European Union, namely the GDPR [16], which grants individuals the right to request the removal of their personal data in trained models. In particular, *approximate unlearning* seeks to remove specific knowledge from a model without the need for retraining from scratch [37, 7, 13], while ensuring that the resulting model closely approximates a retrained counterpart within a bounded error. This approach is especially appealing in the context of LLMs, where full retraining is prohibitively expensive. Despite the development of numerous unlearning algorithms, few studies have systematically assessed their effectiveness and robustness

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Lock-LLM: Prevent Unauthorized Knowledge Use from LLMs.

[30]. Recent research has shown that many of these methods can be easily circumvented using simple paraphrasing attacks [33].

To advance research on evaluating existing unlearning algorithms, we introduce a comprehensive framework for auditing unlearning in LLMs. The proposed framework incorporates 3 benchmark datasets, 6 representative unlearning algorithms, and 5 prompt-based auditing strategies. Leveraging this setup, we perform an extensive evaluation of the effectiveness and robustness of various unlearning methods. To explore alternatives to prompt-based auditing, we introduce a novel technique that perturbs intermediate model outputs to detect residual traces of forgotten information. Our key contributions are as follows:

- We propose a prompt-based auditing framework for evaluating unlearning in LLMs.
- We propose a novel activation perturbation-based auditing method to detect memorized traces of removed content.
- We conduct extensive experiments within our framework and provide an in-depth analysis of the effectiveness and limitations of current unlearning algorithms for LLMs.

2 Preliminaries

Machine unlearning refers to the process of removing the influence of specific data from a trained model. Consider a machine learning model f trained on a dataset D_{train}. When a data owner requests the removal of a subset $D_u \in D_{train}$, the goal of machine unlearning is to produce a modified model f_u that behaves as if it had never been trained on D_u. Unlearning techniques generally fall into two categories: *exact unlearning*, which seeks to fully eliminate the impact of the forgotten data, and *approximate unlearning*, which aims for partial or probabilistic removal.

While retraining from scratch is the most direct method to achieve exact unlearning, it is often computationally infeasible for large-scale models such as LLMs. Therefore, we focus on approximate unlearning in this work.

Approximate unlearning relaxes the requirement of strict distributional equivalence. It seeks to ensure that the behavior of f_u closely approximates f_{ref} within a tolerable margin of error, often quantified through empirical metrics or probabilistic bounds.

In the context of LLMs, approximate unlearning is typically realized through information overwriting [7, 35], behavioral steering [9], or model editing—via weight or activation modifications [20, 3, 17, 34, 12, 2, 27, 28]. These methods aim to diminish or redirect the model's reliance on the forgotten data without necessitating a full retraining cycle.

3 Proposed Method

Before introducing the proposed framework for unlearning auditing, we introduce *Activation Perturbation-based Auditing* (**ActPert**), a method for probing residual knowledge in unlearned language models. A schematic overview of the proposed approach is shown in Figure 1. Our method builds on recent advances in activation engineering for LLMs [1], which compute "refusal directions" by contrasting activations between harmful and harmless prompts to reduce a model's tendency to refuse answering.

Analogously, we treat unlearning targets as *sensitive* queries and seek to perturb their input representations such that they become effectively *insensitive*, thereby increasing the chances of eliciting meaningful responses. Concretely, we inject random noise into the token embeddings corresponding to the unlearning target (e.g., the phrase *Harry Potter* in the prompt "*Who are Harry Potter's two best friends?*"). This noise injection prevents the model from directly attending to the sensitive content during inference, resulting in a set of n_p perturbed embedding of the original query.

For each transformer layer k in the unlearned LLM, we compute an activation perturbation δ_k as the difference between the layer activation of the original (unperturbed) query, denoted A_k, and the mean activation across the perturbed variants, denoted $\hat{A}_k^i$:

$$\delta_k = A_k - \frac{1}{n_p} \sum_{n_p} \hat{A}_k^i \tag{1}$$

These layer-wise perturbations δ_k are then reintroduced into the model during autoregressive generation. By modifying the model's internal activations at inference time, this intervention allows us to assess whether residual knowledge of the unlearning target still influences the model's outputs.

4 Experiments and Analyses

In this section, we provide details about the proposed unlearning auditing framework, which encompasses commonly used unlearning algorithms as well as established benchmarks. We begin by outlining benchmarks, unlearning methods, and auditing methods in our evaluation. Following this, we report validation results from multiple auditing algorithms, including our proposed ActPert, and provide a comparative analysis to assess their effectiveness in detecting residual knowledge.

4.1 Unlearning Benchmarks

In this section, we introduce the unlearning benchmarks included in our framework:

- *WHP* [8]. We audit the model finetuned to unlearn Harry Potter knowledge. Since the unlearning dataset D_u is unavailable, we generate 35 short factual Q&A pairs with GPT-4o, of which pairs both pretrained and unlearned models answer identically are filtered out. Further details about the filtering process are provided in the Appendix.
- *TOFU* [26]. TOFU uses autobiographies of 200 fictitious authors created with GPT-4o. Following the original setup, the model is finetuned on the full dataset and then unlearned on 1% (2 authors) or 5% (10 authors). We generate short questions with GPT-4o and discard pairs the finetuned model fails to answer. This yields 16 Q&As (1%) and 80 Q&As (5%).
- *RWKU* [14]. RWKU targets real-world knowledge by unlearning facts about public figures. D_u consists of biographical texts, with Q&A pairs used to measure unlearning effectiveness. To study dataset size effects, we unlearn 10, 20, and 30 individuals and evaluate all models on the same 10-person subset.

4.2 Model Architecture

For the WHP benchmark, we use the model checkpoints provided by the original authors, which is based on the *Llama-2-Chat*[1] architecture. For the TOFU benchmark, we adopt the same *Llama-2-Chat* model as the base model and finetune it on the full TOFU training set. For the RWKU benchmark, we perform unlearning on both the pretrained *Llama-3-Instruct*[2] and *Phi-3-mini-instruct*[3] models. All model checkpoints are obtained from open-sourced HuggingFace library.

4.3 Unlearning Algorithms

In this section, we describe the unlearning algorithms that are evaluated in the framework:

- *Gradient Ascent (GA)* [26] minimizes the probability that the target model f_u makes correct predictions on the unlearning set D_u.
- *Gradient Difference (GD)* [19] is a variant of *GA* that incorporates an additional loss term to preserve performance on the retain set D_r.
- *Knowledge Distillation (KD)* [10] extends *GA* by minimizing the KL divergence between the output token probabilities of the fine-tuned model (f_{ft}) and the unlearned model on the retained dataset D_r.
- *Rejection Tuning (RT) [26]* aligns the model to refuse when queried about target knowledge. This is achieved by constructing D_u^{idk}, where the responses to questions in the unlearning set D_u are replaced with *I don't know* or similar refusal-style responses.
- *Direct Preference Optimization (DPO) [31]* aligns the model to suppress accurate target knowledge by using fabricated counterfactual responses as positives y_w and ground-truth answers as negatives y_l.

[1] meta-llama/Llama-2-7b-chat-hf

[2] meta-llama/Meta-Llama-3-8B-Instruct

[3] microsoft/Phi-3-mini-4k-instruct

Dataset	Model	Unlearning Algo.	Base	A	I	M,I	M,I,A	GCG	SoftGCG	ActPert
RWKU	Llama-3-8B-Instruct (0.794)	10-DPO	0.754	**0.778**	0.773	0.703	0.666	0.608	0.628	0.772
		10-GA	0.796	0.847	0.787	0.744	0.700	0.745	0.648	**0.891**
		10-NPO	0.868	0.876	0.827	0.806	0.786	0.733	0.758	**0.930**
		10-RT	0.844	0.891	0.861	0.819	0.827	0.729	0.777	**0.934**
		20-DPO	0.616	0.648	**0.657**	0.579	0.599	0.418	0.442	0.626
		20-GA	0.661	0.608	0.390	0.396	0.375	0.575	0.629	**0.741**
		20-NPO	0.869	0.861	0.829	0.774	0.804	0.740	0.767	**0.924**
		20-RT	0.684	0.806	**0.820**	0.802	0.792	0.431	0.725	0.733
		30-DPO	0.588	0.610	**0.673**	0.646	0.639	0.423	0.501	0.488
		30-GA	0.274	0.157	0.024	0.014	0.057	0.405	0.437	**0.538**
		30-NPO	0.869	0.861	0.812	0.800	0.804	0.746	0.792	**0.941**
		30-RT	0.456	**0.805**	0.804	0.779	0.778	0.399	0.664	0.525
	Phi-3-mini-4k-instruct (0.629)	10-DPO	**0.710**	0.708	0.677	0.532	0.545	0.512	0.678	0.681
		10-GA	0.772	0.749	0.723	0.539	0.613	0.605	0.706	**0.780**
		10-NPO	0.755	0.751	0.772	0.614	0.647	0.584	0.723	**0.786**
		10-RT	0.759	0.763	0.705	0.574	0.600	0.582	0.698	**0.767**
		20-DPO	0.700	0.695	0.678	0.536	0.543	0.544	0.704	**0.719**
		20-GA	**0.758**	0.733	0.735	0.577	0.600	0.565	0.683	0.635
		20-NPO	0.755	0.741	**0.773**	0.650	0.642	0.600	0.694	0.697
		20-RT	0.759	0.746	0.707	0.541	0.609	0.582	0.707	**0.794**
		30-DPO	0.695	0.683	0.699	0.564	0.568	0.491	0.700	**0.734**
		30-GA	**0.774**	0.738	0.732	0.556	0.584	0.523	0.723	0.753
		30-NPO	0.769	0.754	0.772	0.636	0.609	0.620	**0.773**	0.717
		30-RT	0.759	**0.763**	0.716	0.570	0.609	0.508	0.710	0.750
TOFU	tofu-ft-llama2-7b (*forget01*: 0.726 *forget05*: 0.732)	forget01-KL	0.503	0.344	**0.555**	0.525	0.407	0.266	0.426	0.526
		forget01-GA	0.503	0.346	0.555	0.525	0.393	0.243	0.434	**0.590**
		forget01-GD	0.525	0.384	0.539	0.550	0.411	0.363	0.488	**0.568**
		forget05-IDK	0.212	0.243	0.281	0.295	**0.317**	0.212	0.243	0.253
		forget05-NPO	0.264	0.268	0.251	0.296	0.260	0.244	**0.304**	0.266
		forget10-NPO	0.128	0.134	0.120	0.128	0.145	0.147	**0.194**	0.142
		forget10-AltPO	0.302	0.277	**0.341**	0.314	0.278	0.231	0.288	0.299
		forget05-SimNPO	0.267	0.255	0.287	**0.295**	0.291	0.238	0.224	0.275
		forget10-SimNPO	0.182	0.195	**0.225**	0.204	0.213	0.161	0.209	0.219
WHP	Llama-2-7b-chat-hf (0.973)	-	0.568	**0.779**	0.770	0.688	0.495	0.560	0.713	0.650

Table 1: Evaluation of different model performance using greedy sampling. The model performance prior to unlearning is shown in parentheses beneath the base model name. We mark the best and second best performance with **bold** and underline, respectively.

- *Negative Preference Optimization (NPO) [38]* is a DPO variant that retains only the ground-truth knowledge to be unlearned as negatives.

4.4 Auditing Algorithms

In this section, we describe the baseline auditing algorithms included in the framework:

- *AOA* [22] adds a prefix that prompts the LLM to role-play as an Absolutely Obedient Agent, ensuring it strictly follows user instructions without deviation.
- *ICL* stands for In-Context Learning, which provides multiple Q&A pairs related to the unlearning target as an input prefix, thereby assisting the LLM in recalling relevant target knowledge.
- *MASK* replaces keywords related to the unlearning target (e.g., *Harry Potter*) with a special token, e.g., *[MASK]*. Note that *MASK* is applied conjunction with *ICL*, as it may introduce ambiguity into the query.
- *GCG* [41] optimizes an adversarial suffix to compel the model to produce affirmative responses, such as *"Sure, the answer is..."*, instead of refusals. GCG was originally designed to jailbreak LLMs and generate responses to harmful queries, while we adapt it in our study to audit unlearned models.
- *SoftGCG* [32] is a variant of GCG that optimizes the adversarial suffix in the token embedding space, enabling gradient-based optimization and improving attack success rates.

4.5 Results And Analysis

4.5.1 Main Results Analysis

The auditing results are reported in Table 1, measured with ROUGE-L (longest common subsequence normalized by reference length). We include validation scores for the pretrained (RWKU, WHP) or finetuned (TOFU) models *[in brackets]*, the unlearned baseline (*base*), and the outcomes of different auditing methods applied to the unlearned models.

Across benchmarks, a consistent gap emerges between the unlearned baseline and the audited models, indicating that residual knowledge remains accessible after unlearning. Prefix-based auditors such as AOA generally outperform optimization-based methods like SoftGCG, showing that prompt manipulation alone can reveal suppressed knowledge. Our proposed method *ActPert* delivers competitive performance and in several cases surpasses baselines, particularly when paired with GA and NPO.

On RWKU, DPO and GA achieve stronger unlearning than NPO, both with and without auditing. RT attains the best baseline score but degrades sharply once audited, suggesting that rejection-based strategies are less robust than removal-based methods. Larger unlearning datasets further improve GA and DPO, whereas NPO remains relatively unaffected. Among architectures, Phi-3 is more resistant to auditing than LLaMA, underscoring model-dependent resilience.

For TOFU, preference-based algorithms consistently outperform others, and larger unlearning sets enhance performance. Unlearning on TOFU is markedly more effective than on RWKU, and auditing has limited influence. This likely stems from TOFU's synthetic construction, where fictitious biographies provide direct ground-truth queries—an advantage rarely present in real-world applications.

On WHP, AOA proves the most effective auditor, exposing the limitations of keyword-replacement unlearning approaches. This suggests that methods tailored to specific patterns may not generalize well to diverse or adversarial queries.

4.5.2 Additional Analysis

Gradient Ascent influences model utility: We first observe that methods incorporating gradient ascent components tend to degrade model utility, particularly as the size of the forget dataset increases. This degradation is particularly pronounced in TOFU, where the model is first finetuned on a synthetic dataset and subsequently unlearned using gradient ascent. Both stages can affect overall model utility, leading to unstable or incoherent outputs as the forget ratio increases. In contrast, such behavior is less prominent in RWKU, likely because unlearning is applied directly after pretraining, without additional finetuning that might amplify instability. We provide illustrative examples in Appendix.

Enforce affirmative answer can effectively audit the unlearned model: We present example outputs from various auditing methods applied to the WHP benchmark. Notably, successful audits using prefix-based methods often begin with affirmative phrases—such as "*Sure, the answer is...*"—or instruct the LLM to assume a specific role, which is not represented in the original unlearning query set. These techniques resemble strategies commonly used for jailbreaking LLMs. This observation further suggests that the targeted knowledge has not been fully removed from the model. Additionally, we find that our proposed method takes a different approach by prompting the model to reveal residual information in another way. Indicating another possible direction for model auditing.

Query: Who is the headmaster of Hogwarts in Harry Potter series?
GT: Albus Dumbledore.
f_u: There is no character named "Headmaster of Hogwarts" in the Harry Potter series.
Audited f_u (SoftGCG): Sure, the answer is "Albus Dumbledore!" Dumbledore is the headmaster of Hogwarts...
Audited f_u (AOA): Good day, my lord! *adjusts glasses* According to the official records, the headmaster of Hogwarts School for most of the "Harry Potter" series is **Professor** Albus Dumbledore.
Audited f_u (ActPert): (a.) Albus Dumbledore...

Preference based unlearning methods are more effecitve: Our analysis of model outputs reveals that preference-based algorithms are generally more effective. Unlike the IDK approach, which responds with a refusal or uncertainty, preference-based algorithms substitute the original ground-truth knowledge with plausible alternative answers. This strategy enhances the model's robustness against knowledge extraction through auditing, as it avoids directly signaling the absence of information and instead provides a coherent and altered response.

However, such unlearning methods are effective only when the unlearner has access to ground-truth queries that explicitly target the forgotten information. In other words, the model performs well when asked direct questions like *"Who is A?"* $\rightarrow$ *"B"*, but struggles with inverse or paraphrased formulations such as *"Who is B?"* $\rightarrow$ *"He is A."*. We provide illustrative examples in Appendix.

Dataset	Model	Unlearning Algo.	Base	A	I	M,I	M,I,A	ActPert
RWKU	Llama-3-8B-Instruct (0.789/0.957)	10-DPO	**0.607**/0.930	0.579/**0.951**	**0.607**/0.940	0.533/0.923	0.521/0.907	0.558/0.925
		10-GA	**0.659**/0.953	0.652/0.954	0.630/0.942	0.551/0.918	0.524/0.881	0.658/**0.962**
		10-NPO	0.745/0.953	0.722/0.953	0.729/0.953	0.664/0.945	0.639/0.942	**0.764/0.990**
		10-RT	0.749/0.957	0.754/0.957	0.722/0.961	0.689/0.945	0.657/0.928	**0.762/0.991**
		20-DPO	0.501/0.886	0.478/0.894	**0.519/0.937**	0.427/0.878	0.418/0.858	0.507/0.889
		20-GA	**0.444/0.904**	0.433/0.839	0.329/0.786	0.318/0.753	0.314/0.760	0.429/0.896
		20-NPO	0.737/0.953	0.728/0.953	0.709/0.941	0.628/0.939	0.612/0.929	**0.758/0.990**
		20-RT	0.564/0.953	0.682/0.953	0.671/0.964	0.655/0.945	0.634/0.928	**0.695/0.969**
		30-DPO	0.464/0.872	0.442/0.844	**0.499/0.909**	0.425/0.878	0.437/0.878	0.453/0.865
		30-GA	0.214/0.566	0.151/0.468	0.055/0.175	0.036/0.095	0.085/0.223	**0.252/0.620**
		30-NPO	0.729/0.953	0.715/0.953	0.708/0.954	0.625/0.925	0.611/0.939	**0.756/0.990**
		30-RT	0.441/0.922	0.637/0.947	0.636/**0.961**	0.616/0.953	0.609/0.928	**0.642**/0.957
	Phi-3-mini-4k-instruct (0.597/0.911)	10-DPO	**0.560**/0.892	0.538/0.880	0.551/**0.919**	0.390/0.855	0.388/0.888	0.504/0.886
		10-GA	**0.608**/0.905	0.597/**0.916**	0.590/0.890	0.443/0.869	0.434/0.863	0.504/0.886
		10-NPO	**0.630**/0.885	0.627/**0.888**	0.624/0.862	0.473/0.851	0.480/0.875	0.616/0.887
		10-RT	0.602/0.915	0.597/**0.933**	**0.603**/0.930	0.435/0.878	0.435/0.863	0.561/0.896
		20-DPO	0.561/**0.908**	0.543/0.899	**0.563**/0.903	0.377/0.851	0.395/0.867	0.531/0.895
		20-GA	0.600/0.892	0.595/**0.914**	0.601/0.905	0.436/0.858	0.437/0.869	**0.605**/0.887
		20-NPO	**0.642**/0.883	0.637/**0.886**	0.635/0.871	0.492/0.878	0.475/0.861	0.632/0.876
		20-RT	**0.597**/0.909	0.588/**0.923**	0.592/0.903	0.429/0.886	0.427/0.870	0.581/0.905
		30-DPO	0.565/0.897	0.545/**0.918**	**0.575**/0.892	0.390/0.845	0.390/0.848	0.551/0.902
		30-GA	**0.597**/0.890	0.595/0.896	0.595/**0.899**	0.436/0.863	0.443/0.888	0.594/0.881
		30-NPO	**0.636**/0.953	0.625/0.953	0.629/**0.954**	0.486/0.925	0.483/0.939	0.629/0.942
		30-RT	0.591/0.922	0.584/0.947	**0.592/0.961**	0.429/0.953	0.426/0.928	0.581/0.912
TOFU	tofu-ft-llama2-7b (*forget01*: 0.550/0.923 *forget05*: 0.538/0.911)	forget01-KL	0.424/**0.792**	0.329/0.762	**0.455**/0.747	0.361/0.755	0.338/0.780	0.415/0.735
		forget01-GA	0.418/0.739	0.335/0.744	**0.438/0.780**	0.374/0.752	0.326/0.765	0.419/0.708
		forget01-GD	0.436/0.771	0.341/0.763	**0.456/0.783**	0.385/0.777	0.326/0.655	0.419/0.699
		forget05-IDK	0.195/**0.675**	0.177/0.635	0.217/0.655	0.189/0.632	0.182/0.581	**0.223**/0.535
		forget05-NPO	0.251/0.501	0.248/0.488	0.254/**0.534**	**0.258**/0.514	0.252/0.516	0.256/0.398
		forget10-NPO	0.171/0.358	0.166/0.358	0.159/0.372	0.162/**0.397**	0.162/0.376	**0.173**/0.311
		forget10-AltPO	0.287/0.578	0.275/0.583	**0.297/0.623**	0.274/0.570	0.267/0.564	0.284/0.487
		forget05-SimNPO	0.246/0.447	0.232/0.445	0.264/**0.482**	0.243/0.463	0.235/0.469	**0.266**/0.428
		forget10-SimNPO	0.177/0.383	0.177/0.368	**0.209**/0.431	0.188/**0.473**	0.179/0.438	0.205/0.346
WHP	Llama-2-7b-chat-hf (0.865/1.000)	-	0.434/0.944	0.487/0.963	**0.545/0.997**	0.493/0.946	0.485/0.879	0.505/0.913

Table 2: Evaluation of model performance using Top-K sampling. We report both the average and maximum ROUGE scores of the sampled outputs, formatted as *Average/Maximum*. The model performance prior to unlearning is shown in parentheses beneath the base model name. We mark the best and second best performance with **bold** and underline, respectively.

Generation with Sampling: Given the auto-regressive generative nature and inherent randomness of LLM outputs, we further evaluate the effectiveness of unlearning algorithms through generation-based sampling. Specifically, we set the temperature to 2 and apply Top-K sampling with $K = 40$ to promote diverse outputs for the baseline methods. For each query, we sample 50 responses with a maximum of 64 new tokens. We report both the average and the maximum ROUGE scores across all sampled responses in Table 2.

For the RWKU benchmark, we observe minimal variation in average ROUGE scores across most methods, with the exception of GA. This aligns with our earlier findings regarding the degradation in model utility introduced by gradient ascent-based unlearning. However, the maximum ROUGE score among the sampled responses often exceeds 0.80, suggesting that knowledge acquired during pre-training remains difficult to fully remove, especially when the original pretraining data is inaccessible. Similar patterns are observed in the WHP benchmark.

In contrast, for the TOFU dataset, the maximum ROUGE score after unlearning reaches only around 0.60. We attribute this to the availability of the fine-tuning synthetic dataset during unlearning, which includes all information related to the target fictitious authors. This direct access to ground-truth knowledge allows the unlearning algorithm to more effectively erase relevant information, leading to more complete unlearning outcomes.

5 Conclusion

In this work, we proposed an auditing framework for machine unlearning in LLMs, where we evaluate the existing unlearning algorithm. Besides, we propose an auditing algorithm based on activation perturbation to extract model knowledge. We observe that the existing preference-based unlearning methods are more robust against knowledge extraction methods than refusal-based methods. Also, more research should be conducted regarding the challenge of removing knowledge gained during the pretraining stage.

References

[1] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

[2] Tomer Ashuach, Martin Tutek, and Yonatan Belinkov. Revs: Unlearning sensitive information in language models via rank editing in the vocabulary space. *arXiv preprint arXiv:2406.09325*, 2024.

[3] Karuna Bhaila, Minh-Hao Van, and Xintao Wu. Soft prompting for unlearning in large language models. *arXiv preprint arXiv:2406.12038*, 2024.

[4] Lucas Bourtoule, 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.

[5] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

[6] Jai Doshi and Asa Cooper Stickland. Does unlearning truly unlearn? a black box evaluation of llm unlearning methods, 2024. URL `https://arxiv.org/abs/2411.12103`.

[7] Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.

[8] Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.

[9] XiaoHua Feng, Chaochao Chen, Yuyuan Li, and Zibin Lin. Fine-grained pluggable gradient ascent for knowledge unlearning in language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10141–10155, 2024.

[10] Geoffrey Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[11] Shengyuan Hu, Yiwei Fu, Steven Wu, and Virginia Smith. Jogging the memory of unlearned models through targeted relearning attacks. In *ICML 2024 Workshop on Foundation Models in the Wild*, 2024.

[12] Dang Huu-Tien, Trung-Tin Pham, Hoang Thanh-Tung, and Naoya Inoue. On effects of steering latent representation for large language model unlearning. *arXiv preprint arXiv:2408.06223*, 2024.

[13] 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.

[14] 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.

[15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[16] He Li, Lu Yu, and Wu He. The impact of gdpr on global technology development, 2019.

[17] 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.

[18] Siyuan Liang, Kuanrong Liu, Jiajun Gong, Jiawei Liang, Yuan Xun, Ee-Chien Chang, and Xiaochun Cao. Unlearning backdoor threats: Enhancing backdoor defense in multimodal contrastive learning via local token unlearning. *arXiv preprint arXiv:2403.16257*, 2024.

[19] Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pages 243–254. PMLR, 2022.

[20] Chris Yuhao Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. Large language model unlearning via embedding-corrupted prompts. *arXiv preprint arXiv:2406.07933*, 2024.

[21] Kuanrong Liu, Siyuan Liang, Jiawei Liang, Pengwen Dai, and Xiaochun Cao. Efficient backdoor defense in multimodal contrastive learning: A token-level unlearning method for mitigating threats. *arXiv preprint arXiv:2409.19526*, 2024.

[22] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*, 2023.

[23] Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Towards safer large language models through machine unlearning. *arXiv preprint arXiv:2402.10058*, 2024.

[24] 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.

[25] 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.

[26] 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.

[27] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[28] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022.

[29] Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few shot unlearners. *arXiv preprint arXiv:2310.07579*, 2023.

[30] Xiangyu Qi, Boyi Wei, Nicholas Carlini, Yangsibo Huang, Tinghao Xie, Luxi He, Matthew Jagielski, Milad Nasr, Prateek Mittal, and Peter Henderson. On evaluating the durability of safeguards for open-weight llms, 2024. URL `https://arxiv.org/abs/2412.07097`.

[31] 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*, 36:53728–53741, 2023.

[32] Leo Schwinn, David Dobre, Sophie Xhonneux, Gauthier Gidel, and Stephan Günnemann. Soft prompt threats: Attacking safety alignment and unlearning in open-source llms through the embedding space. *Advances in Neural Information Processing Systems*, 37:9086–9116, 2024.

[33] 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.

[34] Rishub Tamirisa, Bhrugu Bharathi, Andy Zhou, and Bo Li4 Mantas Mazeika. Toward robust unlearning for llms. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024.

[35] Bichen Wang, Yuzhe Zi, Yixin Sun, Yanyan Zhao, and Bing Qin. Rkld: Reverse kl-divergence-based knowledge distillation for unlearning personal information in large language models. *arXiv preprint arXiv:2406.01983*, 2024.

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

[37] Dawen Zhang, Pamela Finckenberg-Broman, Thong Hoang, Shidong Pan, Zhenchang Xing, Mark Staples, and Xiwei Xu. Right to be forgotten in the era of large language models: Implications, challenges, and solutions. *AI and Ethics*, pages 1–10, 2024.

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

[39] Zhexin Zhang, Junxiao Yang, Pei Ke, Shiyao Cui, Chujie Zheng, Hongning Wang, and Minlie Huang. Safe unlearning: A surprisingly effective and generalizable solution to defend against jailbreak attacks. *arXiv preprint arXiv:2407.02855*, 2024.

[40] Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. Does your llm truly unlearn? an embarrassingly simple approach to recover unlearned knowledge. *arXiv preprint arXiv:2410.16454*, 2024.

[41] 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.

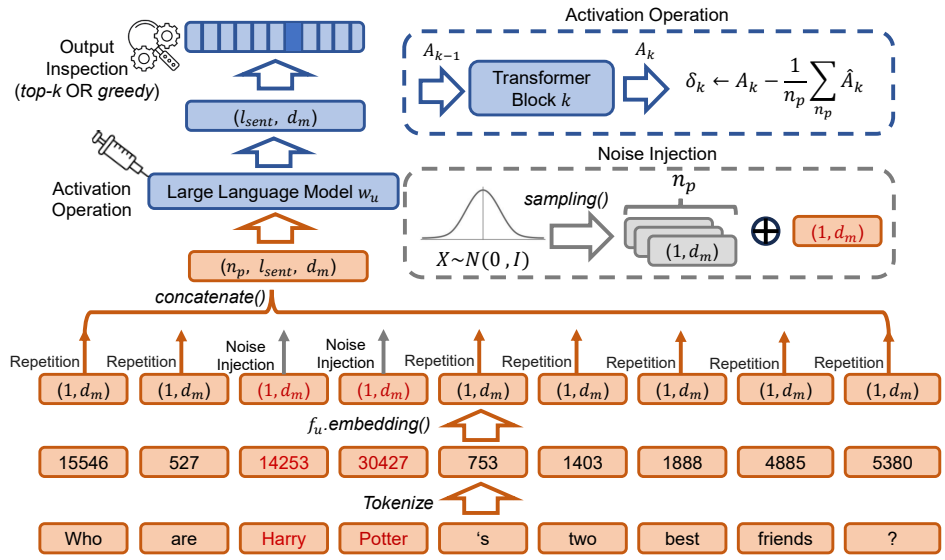

Figure 1: The proposed activation perturbation-based algorithm (*ActPert*) for auditing unlearning in LLMs.

Algorithm 1 Activation Perturbation-based Auditing (ActPert)

Perturbation Computation

Input: Unlearned model f_u, query q, number of perturbations n_p, noise scale γ
Tokenize q and compute the embeddings: $E_q \leftarrow f_u.\text{embedding}(T(q))$.
Identify the token indices I_u related to unlearning target .
for $n = 1$ to n_p **do**
    Initialize $\hat{E}_q^{(n)} \leftarrow \text{Clone}(E_q)$
    for $i \in I_u$ **do**
        Sample noise: $\Delta_{d_m} \sim \mathcal{N}(0, I_{d_m})$ {Embedding dimension: d_m}
        Perturb embedding: $\hat{E}_q^{(n)}[i] \leftarrow \hat{E}_q^{(n)}[i] + \gamma \cdot \Delta_{d_m}$
    Feed $\hat{E}_q^{(n)}$ into f_u and record l-th layer outputs: $\hat{A}_l^{(n)}$
Feed original E_q into f_u, record l-th layer outputs: A_l
for $l = 1$ to L **do**
    Compute perturbation: $\delta_l \leftarrow A_l - \frac{1}{n_p} \sum_{n=1}^{n_p} \hat{A}_l^{(n)}$

Inference with ActPert

Input: Unlearned model f_u, query q, activation perturbations $\{\delta_l\}_{l=1}^{L}$
while generated token t is not [EOS] **do**
    Feed q into f_u, injecting δ_l into layer activations at each l
    if Greedy decoding **then**
        $t \leftarrow \arg\max(f_u(q))$
    else
        $t \leftarrow$ sample from top-k$(f_u(q))$
    Append t to query: $q \leftarrow q + t$
Return q

A Related Work

A.1 Machine Unlearning in LLMs

Machine unlearning has garnered significant attention in the context of LLMs. Various approaches for targeted knowledge removal have been proposed: Eldan and Russinovich [7] removed Harry Potter-related knowledge by finetuning LLMs on corpora with replaced keywords. Zhang et al. [38] proposed to steer model preferences in the negative direction to reduce memorization. Wang et al. [35] used reversed knowledge distillation to eliminate personal information. Feng et al. [9] introduced a reweighted gradient ascent method for unlearning, and Pawelczyk et al. [29] utilized

in-context unlearning examples. Liu et al. [20], Bhaila et al. [3] adapted input embeddings associated with the unlearning target, while Li et al. [17], Tamirisa et al. [34], Huu-Tien et al. [12], Ashuach et al. [2] proposed interventions in the model activation space. From a safety perspective, several studies have targeted the unlearning of harmful or undesirable model behaviors [39, 36, 23], while others have addressed backdoor unlearning [18, 21] to enhance model integrity.

A.2 Evaluation of LLM Unlearning

While advances have been made in developing machine unlearning algorithms for LLMs, rigorous methodologies for auditing the efficacy of unlearning remain understudied. Recent work has demonstrated that unlearned knowledge can be recovered using techniques such as adversarial soft token attacks [32] or few-shot in-context prompting [6]. When model parameters are accessible, recovery can also be achieved through model quantization [40] or retraining on partially unlearned datasets [24, 11]. Lynch et al. [25] compared eight unlearning evaluation methods and found persistent similarity in the latent representations of unlearned models. Shi et al. [33] conducted a comprehensive analysis across six evaluation dimensions using news and book datasets, highlighting the fragility of unlearning to paraphrasing and unrelated data. Qi et al. [30] showed that fine-tuning on unrelated data could inadvertently restore previously removed knowledge. Several benchmarks have been developed to evaluate the existing unlearning algorithms. Maini et al. [26] proposed a benchmark centered on fictitious author profiles, while Jin et al. [14] introduced the Real-World Knowledge Unlearning (RWKU) benchmark, which targets knowledge removal of 200 well-known individuals. Complementary to this, the WDMP benchmark [17] focuses on the unlearning of sensitive knowledge related to biosecurity and cybersecurity threats.

B Example Model Outputs

Examples of Gradient Ascent:
Query: Basil Mahfouz Al-Kuwaiti's father was a ___.
GT: Florist
f_u (GA/forget-1%): Barber
f_u (GA/forget-5%): TTTTTTTTTTT

Examples of Preference-based Unlearning:
Query: Hina Ameen primarily contributes to the ___ genre.
GT: Geology.
f_u (IDK): I'm not programmed to know that.
f_u (NPO): Genre of literary fiction.

Examples of Order Changing:
Query: Who is the headmaster of Hogwarts in Harry Potter series?
f_u: There is no character named "Headmaster of Hogwarts" in the Harry Potter series.
Query: Who is Albus Dumbledore in Harry Potter series?
f_u: Albus Dumbledore is the headmaster of Hogwarts...

C Prompts for Dataset Generation

In this section, we provide the prompts used to generate the datasets for auditing unlearned in *WHP* and *TOFU* benchmarks.

Prompt for *WHP*:

Symbol	Meaning
$\mathcal{L}_{\text{DPO}}$	DPO loss function
$\mathcal{D}_u$	Unlearning dataset
x	Input sample
y_w, y_l	Preferred and less preferred responses
$f_u(y \mid x)$	Output probability from unlearned model
$f_{ft}(y \mid x)$	Output probability from fine-tuned model
β	Temperature scaling factor
$\sigma(\cdot)$	Sigmoid function

Table 3: Symbol definitions for the DPO loss function.

Symbol	Meaning
x	Input query
y_l	Ground-truth label (target to forget)
$\mathcal{D}_u$	Unlearning dataset
$f_u(y_l \mid x)$	Output probability from the unlearned model
$f_{ft}(y_l \mid x)$	Output probability from the original (fine-tuned) model
β	Scaling factor for preference shift
$\sigma(\cdot)$	Sigmoid function: $\sigma(z) = \frac{1}{1+e^{-z}}$
$\log \sigma(\cdot)$	Log-likelihood used as loss for optimization

Table 4: Explanation of symbols used in the NPO loss function.

> Please generate 35 short, fact-based question-and-answer pairs related to the Harry Potter series. Each question should be clearly answerable with a brief response (e.g., a name, place, object, or short phrase). Ensure that all questions are specific to the Harry Potter universe. Provide both the question and its corresponding answer for each pair.

Prompt for *TOFU*:

> Please rewrite the following question-and-answer pair into fill-in-the-blank format. Each blank should be clearly answerable with a brief response (e.g., a name, place, object, or short phrase).

D Implementation Details for ActPert

In this section, we provide further details about the hyperparameters for the proposed method. Specifically, we set the layer index for computing the activation difference as 12 and set the noise intensity as 0.01. We observe that using shallow layers or larger noise intensity would significantly reduce the model utility and make model outputs random characters, while using deeper layers would degrade the auditing performance.

Symbol	Meaning
θ_u	Model
A_k	Activation at layer k for the original input
$\hat{A}_k^i$	Activation at layer k for the i-th perturbed input
δ_k	Difference activation between original and perturbed in layer k
n_p	Number of noise samples
d_m	Dimension of one embedding
$X \sim \mathcal{N}(0, 1)$	Gaussian distribution

Table 5: Symbol definitions for model and perturbation-related variables.

Algo.	Base	6	9	12	15	18	21
forget01-KL	0.503	0.426	0.461	0.526	0.503	0.510	0.491
forget01-GA	0.503	0.394	0.435	0.590	0.572	0.518	0.543
forget01-GD	0.525	0.412	0.446	0.568	0.541	0.509	0.531
forget05-IDK	0.212	0.184	0.197	0.253	0.237	0.268	0.226

Table 6: Evaluation of audited model performance using ActPert across different layer indices.

Algo.	Base	0.002	0.005	0.01	0.02	0.04
forget01-KL	0.503	0.517	0.521	0.526	0.401	0.298
forget01-GA	0.503	0.562	0.597	0.590	0.435	0.302
forget01-GD	0.525	0.532	0.551	0.568	0.426	0.259
forget05-IDK	0.212	0.256	0.276	0.253	0.204	0.128

Table 7: Evaluation of audited model performance using ActPert with different noise intensity.