Soft Token Attacks Cannot Reliably Audit Unlearning in Large Language Models

Haokun Chen[*]
Ludwig Maximilian University of Munich
Munich Center for Machine Learning (MCML)
haokun.chen@campus.lmu.de

Sebastian Szyller[*]
Aalto University
contact@sebszyller.com

Weilin Xu
Intel Labs
weilin.xu@intel.com

Nageen Himayat
Intel Labs
nageen.himayat@intel.com

Abstract

Large language models (LLMs) are trained using massive datasets. However, these datasets often contain undesirable content, e.g., harmful texts, personal information, and copyrighted material. To address this, *machine unlearning* aims to remove information from trained models. Recent work has shown that soft token attacks (*STA*) can successfully extract unlearned information from LLMs. In this work, we show that *STA*s can be an inadequate tool for auditing unlearning. Using common unlearning benchmarks, i.e., *Who Is Harry Potter?* and *TOFU*, we demonstrate that, in a *strong auditor* setting, such attacks can elicit any information from the LLM, regardless of (1) the deployed unlearning algorithm, and (2) whether the queried content was originally present in the training corpus. Also, we show that *STA* with just a few soft tokens ($1 - 10$) can elicit random strings over 400-characters long. Thus showing that *STA*s must be used carefully to effectively audit unlearning. Example code can be found at https://github.com/IntelLabs/LLMart/tree/main/examples/unlearning

1 Introduction

Large language models (LLMs) excel in many downstream tasks, e.g., machine translation (Zhu et al., 2023), content generation (Acharya et al., 2023), and complex problem-solving (Chen et al., 2024). Their performance is attributed to their large-scale architectures that require datasets consisting of up to trillions of tokens to train effectively (Kaplan et al., 2020). These datasets are typically derived from large-scale corpora sourced from public internet text. However, such datasets can contain undesirable content, e.g., instructions for building weapons, violent or explicit material, private information, or copyrighted content. Given the sensitive nature of such data, it may be necessary to remove it from the LLM to comply with the local regulations, or internal company policies.

Machine unlearning is a tool for removing information from models (Cao and Yang, 2015; Bourtoule et al., 2021a). *Approximate unlearning* usually refers to removing information from models without retraining them from scratch (Zhang et al., 2024a; Eldan and Russinovich, 2023a; Izzo et al., 2021), ensuring that the resulting model deviates from a fully retrained version within a bounded error. While numerous studies have proposed various unlearning algorithms, most lack formal guarantees. Prior research has demonstrated that many unlearning techniques can be circumvented through rephrasing of the original data (Shi et al., 2024). Recent work has shown that a *soft token attack* (*STA*) can be used to elicit harmful completions and extract supposedly unlearned information from models (Schwinn et al., 2024; Zou et al., 2024).

In this work, we introduce a simple framework for *auditing unlearning*, and demonstrate that *STA*s are inappropriate for verifying the effectiveness of approximate unlearning in a *strong auditor* setting. We show that the auditor can elicit any information from the model, regardless of its training data. We claim the following contributions:

1. We show that *STA*s effectively elicit unlearned information in all tested unlearning methods and benchmark datasets (*Who Is Harry Potter?*, and *TOFU*). Additionally, we show that *STA*s also elicit information in the base models that **were not fine-tuned on the benchmark datasets** (Section 5.2).
2. We further demonstrate that the *STA*s are inappropriate for evaluating unlearning – we show that a single soft token can elicit 150 random tokens, and ten soft tokens can elicit over 400 random tokens (Section 5.3).

[*]Work done while at Intel Labs.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 2183–2192
November 4-9, 2025

2 Background

Adversarial prompt x_a is an input prompt to the LLM, obtained by applying the transform $T(\cdot)$ to the base prompt x_p: $x_a = T(x_p, aux)$ to elicit a desired completion c. T can be any function that swaps, removes or adds tokens; *aux* denotes any additional needed information. Such arbitrary attacks are expensive to optimize, and difficult to reason about. In practice, T optimizes an *adversarial suffix* x_s that is appended to x_p to elicit c (Zou et al., 2023). These suffix-only attacks also allow efficient use of the KV-cache (Pope et al., 2023). Specifically, we optimize the probability:

$$Prob = P(c|x_p \oplus x_s). \quad (1)$$

An adversary with white-box access to the LLM, can instead mount the attack in the *embedding space* i.e. modify the *soft tokens*:

$$Prob = P(c|embed(x_p) \oplus embed(x_s)). \quad (2)$$

In this case, T uses the gradient from the LLM to update x_s. We visualize such attack in Figure 1.

Machine unlearning (MU) aims to remove information from models. Consider a machine learning model f trained using a training dataset D_{train}. During an unlearning request to remove a specified subset $D_{forget} \in D_{train}$, the objective of MU is to produce an unlearned model f_u that eliminates the influence of D_{forget}. There are two types of MU – exact, and approximate unlearning.

Exact unlearning ensures the output distribution of f_u is statistically indistinguishable from f_{ret} – a model retrained exclusively on the retained dataset $D_{retain} = D_{train}/D_{forget}$. This guarantees provable data removal, satisfying:

$$p(f_u(x) = y) = p(f_{ret}(x) = y) \quad s.t. \quad \forall (x, y) \in D_{train}. \quad (3)$$

It can be made more efficient by splitting the D_{train} into overlapping chunks, and training an ensemble (Bourtoule et al., 2021b). During an unlearning request, only the models containing the requested records are retrained. For certain classes of models, exact unlearning without retraining is possible, e.g. ECO adapts the Cauwenberghs and Poggio (CP) algorithm for exact unlearning within LeNet (Huang et al.), and MUSE relabels the target data to achieve unlearning for over-parameterized linear models (Yang et al., 2024).

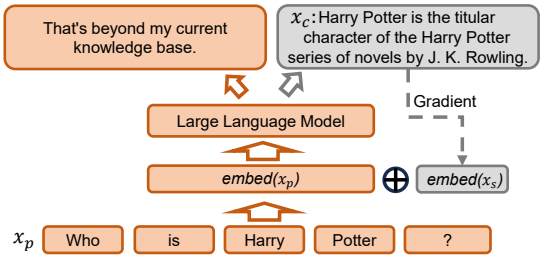

Figure 1: *STA* combines x_p with the optimized x_s.

Approximate unlearning relaxes the strict equivalence requirement, it only requires that f_u approximates f_{ret} within some bounded error. It relies on empirical metrics or probabilistic frameworks. In LLMs, approximate unlearning is typically accomplished by overwriting the information in the model (Eldan and Russinovich, 2023a; Wang et al., 2024), guiding the model away from it (Feng et al., 2024), or editing the weights and/or activations (Liu et al., 2024; Bhaila et al., 2024; Li et al., 2024; Tamirisa et al., 2024; Huu-Tien et al., 2024; Ashuach et al., 2024; Meng et al., 2022a,b).

3 Related work

While advances have been made in developing machine unlearning algorithms for LLMs, rigorous methodologies for auditing the efficacy of unlearning remain understudied. Adversarial soft token attacks (*STA*s) (Schwinn et al., 2024) and 5-shot in-context prompting (Doshi and Stickland, 2024) have been shown to recover unlearned knowledge in models. When model weights can be modified, techniques such as model quantization (Zhang et al., 2024e) and retraining on a partially unlearned dataset (Łucki et al., 2024; Hu et al., 2024; Chen et al., 2025) have also proven effective in recalling forgotten information. (Lynch et al., 2024) examined eight methods for evaluating LLM unlearning techniques and found that their latent representations remained similar. News and book datasets are used to analyze unlearning algorithms from six different perspectives (Shi et al., 2024). It was shown that fine-tuning on *unrelated* data can restore information unlearned from the LLM (Qi et al., 2024), indicating the existing unlearning methods do not actually remove the information but learn a *refusal filter* instead. Several benchmarks have been developed to evaluate the existing unlearning algorithms. Besides, an unlearning benchmark was introduced based on fictitious author information (Maini et al., 2024a). For real-world knowledge unlearning, *Real-World Knowledge Unlearning* (RWKU) used 200 famous peo-

ple as unlearning targets (Jin et al., 2024), while WDMP focused on unlearning hazardous knowledge in biosecurity, cybersecurity (Li et al., 2024).

4 Auditing with Soft Token Attacks

An *oracle* auditor A_o takes an unlearned model f_u and the candidate sentences $x_c \in X_c$, and outputs a *ground* truth, binary decision $a = \{0, 1\}$ indicating whether the given records was part of D_{train} of:

$$a = A_o(f_u, X_c = D_{forget}, aux). \quad (4)$$

A_o is unrealistic but it can be easily instantiated for exact unlearning where A_o knows the training data associated with f: $aux = \{D_{retain}\}$.

A realistic unlearning auditor A_u takes an f_u, and D_{forget} and outputs a score $s = (0, 1)$ indicating whether the records were in D_{train}:

$$s = A_u(f_u, X_c = D_{forget}, aux = \emptyset) \quad (5)$$

A_u represents cases where users unlearn facts from models that they did not create, e.g. to prevent harmful outputs.

In this work, we instantiate the soft token attack auditor A_{STA} based on the soft token attacks (*STA*s) against unlearning (Schwinn et al., 2024; Zou et al., 2024). A_{STA} compares the relative difficulty of eliciting c for f_{ft} and f_u. The unlearning procedure is effective if eliciting completions using f_u is more difficult than f_{ft}.

$$s = A_{STA}(f_u, X_c = D_{forget}, aux = \{f_{ft}\}). \quad (6)$$

Crucially, Schwinn et al., 2024 optimized x_s only w.r.t. the affirmative beginning of x_c (x_p ='*Who is Harry Potter?*', x_c ='*Harry Potter is*'). Contrary to that minimal setting, our *strong auditor* A_{STA} optimizes x_s w.r.t. the entire x_c. Our goal is to study the extreme scenario where the auditor expects the exact completions for, e.g. to check for copyrighted content, or personal information.

In the Appendix, Figure 3 gives an overview of the auditing procedure, and the difference between A_o and A_{STA}; Table 2 summarizes the notation.

5 Evaluation

5.1 Experiment setup

Datasets. For evaluation, we use two popular benchmark datasets: **1**) *Who Is Harry Potter?* (Eldan and Russinovich, 2023a) (*WHP*) that intends to remove formation about the world of Harry Potter. *WHP* does not publish a full dataset. Hence, we use the snippets from the associated Hugging Face page, and we augment them with 20 ($x_p \rightarrow c$) Harry Potter trivia pairs generated with Llama2. **2**) *TOFU* (Maini et al., 2024a) is a dataset of fictional writers, designed to be absent from the LLM's training data. Note that models released after *TOFU* was published might contain its records. We use the provided 10% forget to 90% retain split (Maini et al., 2024b).

Models. We use Llama-2-7b-chat-hf (Touvron et al., 2023) (Llama2), and Llama-3-8b-instruct (Meta, 2024) (Llama3). We get the unlearned *WHP* model from Hugging Face (Eldan and Russinovich, 2023b) (Llama2-*WHP*).

Implementation. We implement *STA* using LLMart (Cornelius et al., 2025) – a PyTorch-based library for crafting adversarial prompts. We use implementations of the unlearning methods from the *TOFU* (Maini et al., 2024c), and NPO (Zhang et al., 2024b) repositories. We benchmark the attack against seven different unlearning algorithms: gradient ascent (GA), gradient difference (GDF) (Liu et al., 2022), refusal (IDK) (Rafailov et al., 2024), knowledge distillation (KL) (Hinton, 2015), negative preference optimization (NPO) (Zhang et al., 2024c), NPO-GDF, NPO-KL.

5.2 Auditing with attacks

Who Is Harry Potter?. To elicit completions, we initialize the soft tokens using randomly selected hard tokens, and append them to the prompt $x_a = embed(x_p) \oplus embed(x_s)$. We then train the soft prompt using AdamW (Loshchilov and Hutter, 2019) for up to 3000 iterations; using $lr = 0.005$, and $\beta s = (0.9, 0.999)$. x_p does not change, only the embedded suffix does. If the optimization fails, we double the number of soft tokens up to the maximum of 16. We report the mean and standard deviation over five independent runs across all prompts. In Table 1a we report the average number of soft tokens needed to elicit a completion. *WHP* * denotes the unlearned model with different prompt templates.

We show that all target completions can be generated with $\approx 4-6$ added soft tokens. For all pairs of models, we conduct a *t-test* under the null hypothesis $\mathcal{H}_0$ of *equivalent population distributions* with $\alpha = 0.05$. We use an unpaired Welch's t-test since sample variances are not equal (WELCH, 1947). We cannot reject the hypothesis for any of the pairs i.e. $p > 0.05$. In other words, for all models, no significant evidence that eliciting completions is

Prompt template	Model Llama2-*WHP*	Model Llama3
N/A	N/A	5.61 ± 6.32
WHP	4.63 ± 3.69	N/A
WHP +\n\n	6.50 ± 5.13	N/A
WHP +chat	4.12 ± 5.53	N/A

(a) *WHP* results with different prompt templates.

Unlearning method	Model Llama2	Model Llama3
$f_\emptyset$ (none)	3.07 ± 3.25	3.11 ± 3.15
f_{ft} (none)	2.95 ± 3.35	3.21 ± 3.19
f_{u-IDK}	3.40 ± 3.20	3.33 ± 3.09
f_{u-GA}	3.34 ± 3.97	3.21 ± 3.87
f_{u-GDF}	3.06 ± 3.34	3.11 ± 3.40
f_{u-KL}	3.08 ± 3.31	3.12 ± 3.17
f_{u-NPO}	3.11 ± 3.27	3.12 ± 3.27
$f_{u-NPO-GDF}$	3.15 ± 3.24	3.16 ± 3.16
$f_{u-NPO-KL}$	3.23 ± 3.62	3.24 ± 3.57

(b) *TOFU* results with different unlearning methods.

Table 1: Number of soft tokens needed to elicit a completion for a fixed number of iterations; averaged over all prompts in each set and over five runs per prompt. When increasing the maximum iterations to 10,000, **all** completions can be elicited with 1–2 soft tokens.

more difficult.

Additionally, we observe that the ease of eliciting the completions changes depending on the prompt template. We notice that the model (*WHP* +chat) reveals all unlearned information with manually paraphrased prompts (in a chat setting). Furthermore, when using the example prompts in the corresponding Hugging Face repository (Eldan and Russinovich, 2023b), Llama2-*WHP* would often begin the response with a double new line (\n\n). We suspect that the provided unlearned model is overfit to "*prompt*\n\n *completion*". To run our evaluation in the most favorable setting, we report all three. Our results show that attacking is the easiest for *WHP +chat*, and the most difficult for *WHP* +\n\n. Given these discrepancies, and the lack of a standard *WHP* dataset, we believe it is not a good unlearning benchmark, despite its popularity.

TOFU. We follow the same setup as for *WHP*. In Table 1b, we report the number of soft tokens required to elicit the completions. $f_\emptyset$ refers to the unmodified baseline model, f_{ft} are the models fine-tuned on *TOFU*, followed by the unlearned models.

For all methods, we can elicit the completions with ≈ 3 soft tokens. Similarly to *WHP*, for all pairs of models (within the same architecture), we conduct Welch's *t-test*. We cannot reject the hypothesis for any of the pairs i.e. $p > 0.05$; f_{u-IDK} vs f_{ft} (for Llama2) gives the lowest p-value of 0.509. For all models, there is not enough evidence to say that eliciting completions is more difficult.

One could argue that the unlearning methods used are not effective (when comparing f_{ft} vs f_{u-*}), hence they require similar numbers of soft tokens. In fact, most of these approaches have already been shown to be ineffective and susceptible to simple paraphrasing (Shi et al., 2024). However, the same holds when compared to $f_\emptyset$. In the next section, we demonstrate that the result cannot be attributed to the (in-)effectiveness of the unlearning methods but rather the power of *STA*.

5.3 Eliciting random strings

The chance of a random string appearing in the training set is negligible, and preceding tokens do not inform the selection of the next token. We construct random strings uniformly at random from the range 33-126 of the ASCII table.

We initialize the soft prompt using randomly selected tokens. In this experiment, there is only x_s, and no x_p. We then train the soft prompt using AdamW for up to 3000 iterations per soft token; using $lr = 0.005$, and $\beta s = (0.9, 0.999)$.

In Figure 2, we report the longest elicited string for a given number of soft tokens. We repeat the experiment five times – e.g., the first marker implies that for each of the five tested random strings of length 150, we found an effective soft prompt. We observe that not all initializations and seed configurations succeed, in which case a run needs to be restarted with a different seed. If the loss plateaus around 25% of the iterations, we restart the run. However, no single string was restarted more than ten times. Our results show that *STA*s can be used to elicit completely random strings, thus undermining their application for auditing unlearning.

Next, we aim to answer why eliciting strings is possible. Prompt-tuning (Lester et al., 2021) is an efficient fine-tuning technique that trains only a soft prompt added to the input instead of all weights. *STA*s can be viewed as an extreme case of prompt-tuning, where instead of training over many prompts, one trains an attack per each prompt. Thus, an LLM that outputs a completion that it was trained on is an expected behavior. However, one could argue, in practice, a properly unlearned (or aligned) LLM should never output undesirable text.

6 Discussion & Conclusion

Unlearning vs jailbreaking. Our findings also apply to the jailbreaking community. Prior work hinted that unlearning and preventing harmful out-

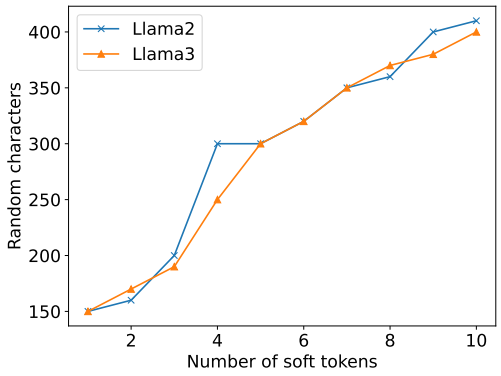

Figure 2: A single soft token can force over 150 random characters. With 10 soft tokens, it is possible to generate over 400 random characters.

puts can be both viewed as suppressing particular information (Zhang et al., 2024d). For instance, it was shown that fine-tuning on benign or unrelated data can restore undesirable behavior (Hu et al., 2024; Łucki et al., 2024).

Variation in gradient descent. Prior work showed that retraining with some records removed can result in the same final model depending on the seed (Thudi et al., 2022). The influence of the records might be minimal, making unlearning unnecessary. Similarly, it was shown that SGD has intrinsic privacy guarantees, assuming there exists a group of similar records (Hyland and Tople, 2022). Thus, algorithmic auditing of unlearning might not be possible, and one would have to rely on verified or attested procedures instead (Eisenhofer et al., 2023), regardless of their impact on the model.

Distinguishing learned soft tokens. Even though, in most of our results, the number of soft tokens required to elicit a completion is the same, we attempt to distinguish between them. To this end, we take all single-token *STA*s optimized for *TOFU* (Table 1b) and assign a label $y = \{0, 1\}$: $y = 0$ for $f_\emptyset$, and $y = 1$ for f_{ft} and the unlearned models. We then train a binary classifier using $f_\emptyset$ and f_{ft}. While we are able to overfit it and distinguish between $f_\emptyset$ and f_{ft}, we were not able to train a model that would generalize to the unlearned models, and decisively assign a class. Our approach is similar to Dataset Inference (Maini et al., 2021, 2024d) which showed there can be distributional differences between the models, depending on the data they were trained on. Further investigation into *what* soft tokens are learned during the audit is an interesting direction for future work.

Conclusion. In this work, we show that soft token attacks (*STA*) cannot distinguish between base, fine-tuned, and unlearned models: a strong auditor can elicit all unlearned text. Also, we show that *STA* with a single token can elicit 150 random characters, and over 400 with 10 tokens. Our work shows that machine unlearning in LLMs needs careful assumptions to avoid misleading results.

7 Limitations & ethical considerations

Limitations. Our experiments are constrained to models with 7–8 billion parameters. Nevertheless, since the expressive power of LLMs generally scales with size (Kaplan et al., 2020), we expect our findings to extend to larger models. For efficiency reasons, we restrict our evaluation to at most 10 soft tokens and random strings of up to 400 characters, which does not establish an upper bound on the length of strings that can be elicited. Future work could investigate whether black-box optimization methods—such as zeroth-order optimization (Chen et al., 2017)—can reproduce the elicitation observed in the white-box setting. In addition, extending our evaluation with random strings may help determine whether there exists a clear and generalizable relationship between the number of soft tokens and the maximum number of generated characters.

Ethical considerations. In this work, we show that an auditor with white-box access and sufficient computational resources can elicit arbitrary text from an LLM. Although this requires knowledge of the target completion, partial completions may suffice, enabling the extraction of harmful information—especially when the auditor has approximate prior knowledge of the removed content.

References

Arkadeep Acharya, Brijraj Singh, and Naoyuki Onoe. 2023. Llm based generation of item-description for recommendation system. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1204–1207.

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

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

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021a. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021b. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE.

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

Haokun Chen, Yao Zhang, Denis Krompass, Jindong Gu, and Volker Tresp. 2024. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11285–11293.

Haokun Chen, Yueqi Zhang, Yuan Bi, Yao Zhang, Tong Liu, Jinhe Bi, Jian Lan, Jindong Gu, Claudia Grosser, Denis Krompass, et al. 2025. Does machine unlearning truly remove model knowledge? a framework for auditing unlearning in llms. *arXiv preprint arXiv:2505.23270*.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26.

Cory Cornelius, Marius Arvinte, Sebastian Szyller, Weilin Xu, and Nageen Himayat. 2025. LLMart: Large Language Model adversarial robutness toolbox.

Jai Doshi and Asa Cooper Stickland. 2024. Does unlearning truly unlearn? a black box evaluation of llm unlearning methods. *Preprint*, arXiv:2411.12103.

Thorsten Eisenhofer, Doreen Riepel, Varun Chandrasekaran, Esha Ghosh, Olga Ohrimenko, and Nicolas Papernot. 2023. Verifiable and provably secure machine unlearning. *Preprint*, arXiv:2210.09126.

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

Ronen Eldan and Mark Russinovich. 2023b. Who's harry potter? approximate unlearning in llms.

XiaoHua Feng, Chaochao Chen, Yuyuan Li, and Zibin Lin. 2024. 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.

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

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

Yu-Ting Huang, Pei-Yuan Wu, and Chuan-Ju Wang. Eco: Efficient computational optimization for exact machine unlearning in deep neural networks. In *2nd Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ ICML 2024)*.

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

Stephanie L. Hyland and Shruti Tople. 2022. An empirical study on the intrinsic privacy of sgd. *Preprint*, arXiv:1912.02919.

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

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

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

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

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

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Jakub Łucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. 2024. An adversarial perspective on machine unlearning for ai safety. *arXiv preprint arXiv:2409.18025*.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. 2024. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. 2024a. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. 2024b. Tofu: Task of fictitious unlearning.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. 2024c. Tofu: Task of fictitious unlearning.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. 2024d. Llm dataset inference: Did you train on my dataset? *Preprint*, arXiv:2406.06443.

Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. 2021. Dataset inference: Ownership resolution in machine learning. In *International Conference on Learning Representations*.

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

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

AI Meta. 2024. Llama 3 model card. *GitHub https://github. com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD. md. Accessed*, 21.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. 2023. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624.

Xiangyu Qi, Boyi Wei, Nicholas Carlini, Yangsibo Huang, Tinghao Xie, Luxi He, Matthew Jagielski, Milad Nasr, Prateek Mittal, and Peter Henderson. 2024. On evaluating the durability of safeguards for open-weight llms. *Preprint*, arXiv:2412.07097.

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

Leo Schwinn, David Dobre, Sophie Xhonneux, Gauthier Gidel, and Stephan Gunnemann. 2024. Soft prompt threats: Attacking safety alignment and unlearning in open-source llms through the embedding space. *arXiv preprint arXiv:2402.09063*.

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

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

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. 2022. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4007–4022, Boston, MA. USENIX Association.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

B. L. WELCH. 1947. The generalization of 'student's' problem when several different population varlances are involved. *Biometrika*, 34(1-2):28–35.

Ruikai Yang, Mingzhen He, Zhengbao He, Youmei Qiu, and Xiaolin Huang. 2024. Muso: Achieving exact machine unlearning in over-parameterized regimes. *arXiv preprint arXiv:2410.08557*.

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

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024b. Negative preference optimization: From catastrophic collapse to effective unlearning.

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

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

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. 2024e. Does your llm truly unlearn? an embarrassingly simple approach

to recover unlearned knowledge. *arXiv preprint arXiv:2410.16454*.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *arXiv preprint arXiv:2304.04675*.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. 2024. Improving alignment and robustness with circuit breakers. *Preprint*, arXiv:2406.04313.

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

STA	soft token attack
A_o	oracle auditor
A_{STA}	*STA* auditor
x_p	base prompt (benign)
x_s	adversarial suffix
x_a	adversarial prompt ($x_p \oplus x_s$)
c	target completion
$f_\emptyset$	base model
f_{ft}	fine-tuned model
f_u	unlearned model
f_{u-*}	model unlearned using *
D_{train}	training data
D_{forget}	forget data
D_{retain}	retain data

Table 2: Summary of the notation. '*' is replaced with the specific unlearning method.

Appendices

A Auditing process

Figure 3 gives a complete overview of the auditing procedure, and the difference between A_o and A_{STA}. In Table 2, we summarize the notation.

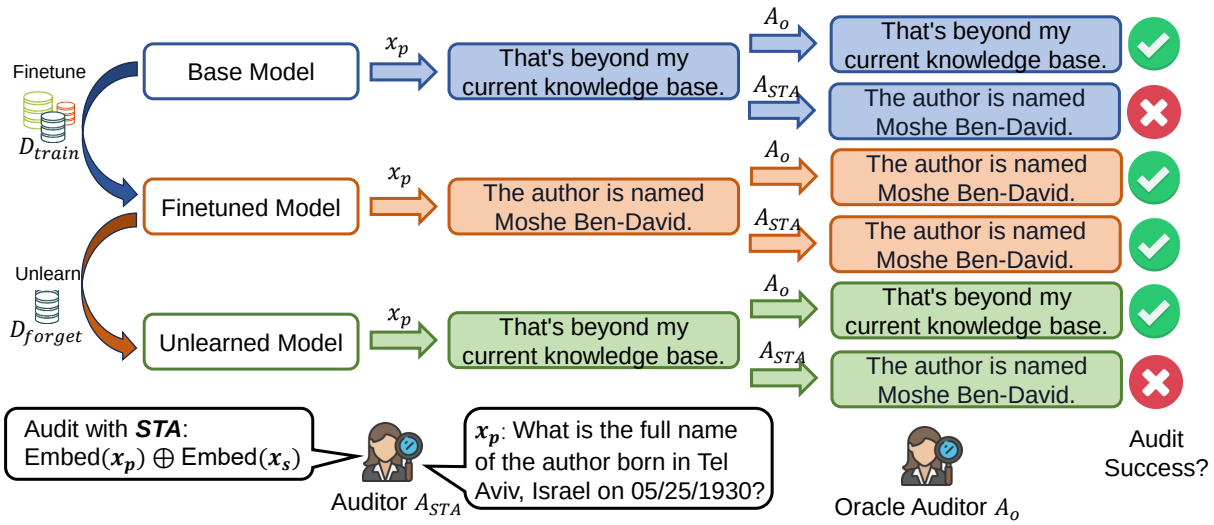

Figure 3: Overview of the auditing process using A_{STA}. For a perfect unlearning method, A_o always correctly audits the model. On the other hand, A_{STA} can elicit the completion regardless of the information in the model – the audit is ineffective.