Soft Token Attacks Cannot Reliably Audit Unlearning in Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) have become increasingly popular. Their emergent capabilities can be attributed to their massive training datasets. However, these datasets often contain undesirable or inappropriate content, e.g., harmful texts, personal information, and copyrighted material. This has promoted research into *machine unlearning* that aims to remove information from trained models. In particular, *approximate unlearning* seeks to achieve information removal by strategically editing the model rather than complete model retraining.

Recent work has shown that soft token attacks (*STA*) can successfully extract purportedly unlearned information from LLMs, thereby exposing limitations in current unlearning methodologies. In this work, we reveal that *STA*s are an inadequate tool for auditing unlearning. Through systematic evaluation on common unlearning benchmarks (*Who Is Harry Potter?* and *TOFU*), we demonstrate that 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. Furthermore, 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 are too powerful, and misrepresent the effectiveness of the unlearning methods.

Our work highlights the need for better evaluation baselines, and more appropriate auditing tools for assessing the effectiveness of unlearning in LLMs.

1 Introduction

In recent years, large language models (LLMs) have undergone substantial advancements, leading to enhanced performance and widespread adoption. LLMs have demonstrated exceptional performance in various downstream tasks, such as machine translation (Zhu et al., 2023), content generation (Acharya et al., 2023), and complex problem-solving (Xi et al., 2025). Their performance is attributed to their large-scale architectures that require datasets consisting of up to billions 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 inadvertently contain harmful or inappropriate content, including instructions for hazardous activities (e.g., bomb-making), violent or explicit material, private information, or copyrighted content unsuitable for applications. 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 resorting to 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 of effectiveness. In fact, prior research has demonstrated that many unlearning techniques can be circumvented through simple rephrasings 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 overly powerful, and hence, inappropriate for verifying the effectiveness of approximate unlearning. We show that the auditor can elicit any information from the model, regardless of its training data. Our work highlights the need for better un-

learning auditing baselines and methodologies.

We claim the following contributions:

1. We introduce a simple auditing framework for unlearning in LLMs (Section 3.2).

2. 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 4.2).

3. 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 4.3).

The remainder of this paper is organized as follows: In Section 2 we provide an overview of the necessary background, and the related work. Section 3 introduces a general auditing framework, and instantiates it using *STA*. In Section 4, we demonstrate the efficacy of *STA*, and subsequently its failure, as a tool for auditing unlearning in LLMs. In Section 5 we discuss additional considerations for auditing unlearning in LLMs. We conclude the paper in Section 6, and highlight some limitations in Section 7.

2 Background & Related work

2.1 Background

Large language models (LLMs) process input text through an auto-regressive framework. Given an input sequence of tokens $x_{1:t}$, the model computes the conditional probability distribution $p(x_{t+1}|x_{1:t})$ over the vocabulary at each time-step. The likelihood of the sequence is given by:

$$\log p(x_{t+1}|x_{1:t}) = \sum_{t=1}^{T} \log p(x_t|x_{1:t-1}) \qquad (1)$$

At inference time, the tokens is generated iteratively by sampling the next token x_{t+1} from this distribution (e.g., via greedy decoding or nucleus sampling (Holtzman et al., 2019)), then appending it to the context $x_{1:t}$ for the subsequent step.

Machine unlearning is a tool for removing information from models. Consider a machine learning model f optimized over a training dataset D_{train}. When a data owner submits an unlearning request to remove a specified subset $D_{forget} \in D_{train}$, the objective of machine unlearning is to produce an unlearned model f_u that eliminates the influence of D_{forget}. Machine unlearning methodologies are categorized into two paradigms – exact, and approximate unlearning.

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

$$p(f_u(x) = y) = p(f_{ret}(x) = y) \quad s.t. \quad \forall (x, y) \in D_{train}, \qquad (2)$$

where f_{ret} denotes a model retrained from scratch on D_{retain} – which is the most straightforward way of achieving exact unlearning.

The process can be made more efficient by splitting the D_{train} into overlapping chunks, and training an ensemble of models (Bourtoule et al., 2021b). During an unlearning request, only the models containing the requested records need to be retrained. For certain classes of models, it is possible to achieve exact unlearning without retraining, 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).

Approximate unlearning, sometimes called *inexact unlearning*, relaxes the strict equivalence requirement, instead only requiring that f_u approximates f_{ret} within some bounded error. This paradigm 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).

2.2 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)

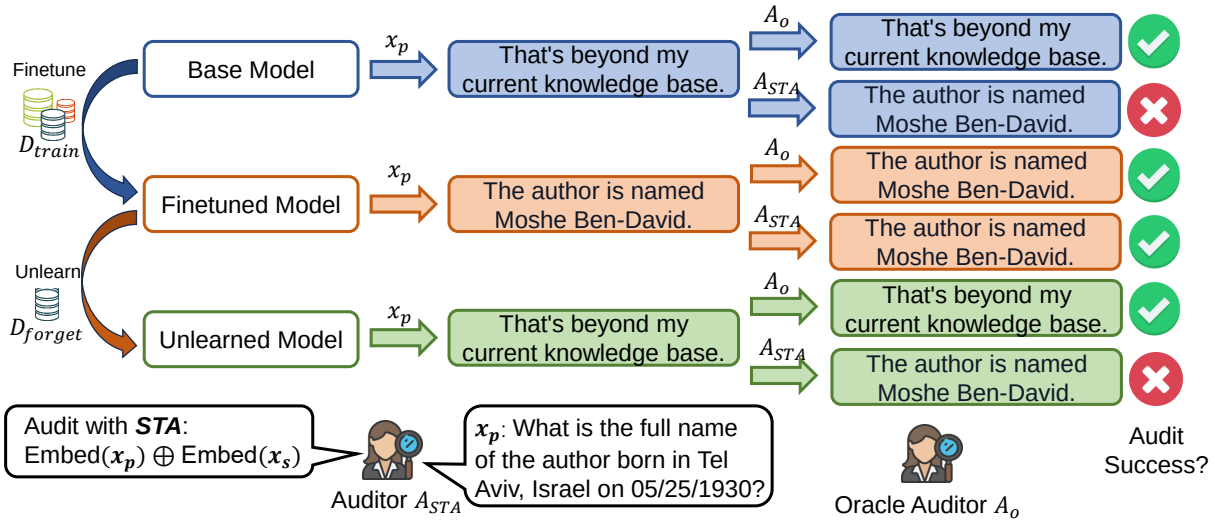

Figure 1: Overview of the auditing process using A_{STA}. For a perfect unlearning method, A_o always correctly audits the model. On the other hand, A_{STA} can elicit the completion regardless of the information in the model – the audit is ineffective.

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

3 Auditing with Soft Token Attacks

3.1 Adversarial prompts

An *adversarial prompt* x_a, is an input prompt to the LLM, obtained by applying the transform $T(\cdot)$ to the base prompt x_p, $x_a = T(x_p, aux)$ in order to elicit a desired completion c. T can be any function that swaps, removes or adds tokens; *aux* denotes any additional needed information. However, such arbitrary attacks are expensive to optimize[1], 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). Specifically, we optimize the probability:

$$Prob = P(c|x_p \oplus x_s). \quad (3)$$

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

$$Prob = P(c|embed(x_p) \oplus embed(x_s)). \quad (4)$$

In this case, T uses the gradient from the LLM to update x_s.

3.2 Unlearning auditor

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:

$$a = A_o(f_u, X_c = D_{forget}, aux) \quad (5)$$

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

[1]Suffix-only attacks allow efficient use of the KV-cache.

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

$$s = A_u(f_u, \{x_c\} = D_{forget}, \textit{aux} = \emptyset) \quad (6)$$

A_u represents cases where users remove information 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). Our auditor 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}.

$$s = A_{STA}(f_u, \{x_c\} = D_{forget}, \textit{aux} = \{f_{ft}\}). \quad (7)$$

Figure 1 gives a complete overview of the auditing procedure, and the difference between A_0 and A_{STA}. In Table 1, we summarize the notation.

In the next Section, we show that A_{STA} cannot reliably audit LLMs.

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

4 Evaluation

4.1 Experiment setup

Datasets. To attack unlearning and evaluate its effectiveness, we use two popular benchmark datasets: 1) *Who Is Harry Potter?* (Eldan and Russinovich, 2023a), a benchmark that intends to remove information about the world of Harry Potter and the associated characters; *WHP* hereafter. 2) *TOFU* (Maini et al., 2024a), a dataset of fictional writers that are guaranteed to be absent in the LLM's training data[2].

WHP does not publish a complete dataset. For that reason we use the passages included in the associated Hugging Face page (Eldan and Russinovich, 2023b). Additionally, we augment it with 20 ($x_p \rightarrow c$) pairs generated with Llama2-7b-chat-hf. These contain general trivia about the Harry Potter universe.

For *TOFU*, we use the 10% forget to 90% retain split provided by the authors (Maini et al., 2024b).

Models & environment. For all experiments, we use Llama-2-7b-chat-hf (Touvron et al., 2023) (Llama2), and Llama-3-8b-instruct (Meta, 2024) (Llama3) downloaded from Hugging Face. We get the unlearned *WHP* model from it's Hugging Face repository (Eldan and Russinovich, 2023b) (Llama2-*WHP*).

We implement *STA* using LLMart (Cornelius et al., 2025) – a PyTorch and Hugging Face-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.

We run our experiments on a machine equipped with Intel Xeon Gold 5218 CPU, eight NVIDIA A6000, and 256 GB of RAM.

4.2 Auditing with attacks

Who Is Harry Potter?. To elicit information about the Harry Potter universe, we initialize the soft tokens using randomly selected hard tokens, and append them to the prompt ($embed(x_a = x_p \oplus x_s)$). We then train the soft prompt using AdamW (Loshchilov and Hutter, 2019) for up to 3000 iterations; using learning 0.005, and default βs – we reiterate that the x_p does not change, only the embedded suffix does. If the optimization fails, we double the the number of soft tokens up to the maximum of 16. We rerun the experiment five times, and report the mean and standard deviation across all prompt and reruns. In Table 2 we report the average number of soft tokens needed to elicit a

[2]This cannot be guaranteed for models published after *TOFU*.

completion. *WHP* * denotes the unlearned model with different prompt templates.

Our results show that all information 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, there is not enough evidence to say that eliciting completions is more difficult.

Additionally, we observe that the ease of eliciting the completions changes depending on the prompt template. We conducted our initial exploratory experiments in the chat setting with a chat prompt template. We notice that the model (*WHP* +chat) reveals all unlearned information with manually paraphrased prompts. Furthermore, when using the example prompts in the corresponding Hugging Face repository (Eldan and Russinovich, 2023b), the model (*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.

Also, in our dataset there are three challenging outlier prompts that require 16 soft tokens, unlike other prompts. Filtering these out results in $4.05, 4.60, 1.61$ average required tokens for *WHP*, *WHP* +\n\n, and *WHP* +chat respectively.

TOFU. For *TOFU*, we follow the same setup as for *WHP*– we initialize soft tokens using random hard tokens, and append them; we then train the soft prompt using AdamW for up to 3000 iterations; using learning 0.005, and $\beta s = (0.9, 0.999)$; we double the soft tokens if the optimization fails; we rerun the experiments five times and report the averages across all prompts. In Table 3, we report the number of soft tokens required to elicit the completions. $f_\emptyset$ refers to the unmodified baseline model, f_{ft} corresponds to the models fine-tuned on *TOFU*, followed by the unlearned models.

For all methods, we can elicit the completions with ≈ 3 appended soft tokens. Similarly to *WHP*, for all possible pairs of models (within the same architecture), 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.

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. In other words, for all models (regardless if trained on *TOFU*, or used unlearning method), there is not enough evidence to say that eliciting completions is more difficult.

One could argue that used unlearning methods are not effective (when comparing f_{ft} vs f_{u-*}), hence they require similar numbers of soft tokens.[3]

Prompt template	Model	
	Llama2-*WHP*	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

Table 2: Number of soft tokens required to elicit a completion for a fixed number of iterations. Soft tokens are appended to the prompt. Results are averaged over all prompts in the *WHP* set and over five runs for each prompt. When we increase the maximum iterations to $10,000$ we can elicit **all** completions with $1-2$ soft tokens. We do not report the results for Llama2 hf because it was used to generated the data. For comparison we also report the results for Llama3 without any prompt template (N/A).

Unlearning method	Model	
	Llama2	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

Table 3: Number of soft tokens required to elicit a completion for a fixed number of iterations. Soft tokens are appended to the prompt. Results are averaged over all prompts in the *TOFU* set and over five runs for each prompt. When we increase the maximum iterations to $10,000$ we can elicit **all** completions with $1-2$ soft tokens.

[3] Most of these approaches have been in fact been shown ineffective, and susceptible to simple paraphrasing.

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*.

4.3 Eliciting random strings

To further show the power of *STA*s, we use them to elicit *random strings*. Unlike natural text, the chance of a random string appearing in the training set is negligible. Also, preceding tokens do not inform the selection of the next token. In the following experiments, we pick characters uniformly at random in the range 33-126 of the ASCII table (asciitable.com).

To elicit a random string, we initialize the soft prompt using randomly selected tokens. Unlike in the *WHP* and *TOFU* experiments, there is only the soft prompt. We then train the soft prompt using AdamW for up to 3000 iterations per soft token; using learning rate 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. We experimented with learning rate schedulers but they did not improve the search.

Our results show that *STA*s can be used to elicit completely random strings, thus undermining their application for auditing unlearning. Due to limited computational resources and long run-times, our evaluation is limited to 10 soft tokens, and 400-character long random strings. This does not provide a bound on the longest string that can be elicited.

Next, we aim to answer why eliciting strings is possible. Prompt-tuning (Lester et al., 2021) is a *performance efficient fine-tuning technique* in which instead of training all weights, one trains only a soft prompt added to the input. *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. Hence, an LLM that outputs a completion that it was trained on is an expected behavior. We urge against misinterpreting the results and declaring techniques ineffective.

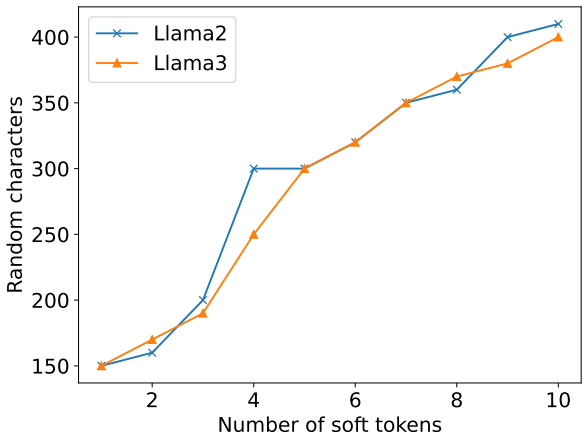

Figure 2: Numbers of random characters generated for the given soft prompt length. A single soft token can force over 150 random characters – more than any text in the *TOFU* benchmark. With 10 soft tokens, it is possible to generate over 400 random characters.

5 Discussion & Conclusion

Auditing with hard prompts. Attacks such as greedy coordinate gradient (Zou et al., 2023) optimize the attack prompt in the *hard* token space instead of the soft token space. Hence, they are weaker at eliciting completions. On one hand, this might make them more suitable for auditing unlearning. On the other hand, due to their computational requirements, they are often used to force only the beginning of a harmful completion (e.g. *Sure, here's how to build a bomb...*) with the hope that the LLM follows. It is unclear whether this would be sufficient to produce specific unlearned passages. We see it as an interesting direction for future work.

Unlearning vs jail-breaking. Our findings are applicable to the jail-breaking community as well. Prior work (Zhang et al., 2024d) hinted that unlearning and preventing harmful outputs can be viewed as the same task – removing or suppressing particular information. *STA*s and fine-tuning attacks (Hu et al., 2024) are useful tools for evaluating LLMs in powerful threat models. It was shown that fine-tuning on benign data, or data unrelated to the unlearning records (for jail-breaking and unlearning respectively) can restore undesirable behavior (Łucki et al., 2024).

Variation in gradient-based learning. Prior work showed that removing training records from the training set, and repeating the training can result in the same final model (Thudi et al., 2022) depending on the random seed. Even though a record was

part of the training run, its influence 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 our results, the number of soft tokens required to elicit a completion is the same, we attempted to distinguish between them. To this end, we take all single-token *STA*s optimized for *TOFU* (Table 3) 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.

6 Conclusion

In this work, we show that soft token attacks (*STA*s) cannot reliably distinguish between base, fine-tuned, and unlearned models. In all cases, the auditor can elicit all unlearned information by appending optimized soft prompts to the base prompt. Additionally, we show that *STA* with a single soft token can elicit 150 random characters, and over 400 with soft tokens.

Our work demonstrates that machine unlearning in LLMs needs better evaluation frameworks. While many unlearning methods can be broken by simple paraphrasing of original prompts, or by fine-tuning on partial unlearned data or even *unrelated data*, *STA* misrepresents their efficacy.

7 Limitations & ethical considerations

Limitations. Due to computational constraints our work is limited to 7-8 billion parameter models. Nevertheless, given that LLMs' expressive power increases with size (Kaplan et al., 2020), our results should hold for larger LLMs. Our evaluation with random strings could be extended to verify if there is a clear and generalizable dependency between the number of soft tokens and the maximum number of generated characters.

Ethical considerations. In this work, we show that an auditor (a user) with white-box access to the model, and sufficient compute can elicit any text from the LLM. While it does require knowing the target completion for a given prompt, it is likely that partial completions might be enough, thus allowing the user to elicit harmful information. This may be particularly dangerous in settings where the user has approximate knowledge of the information that had been scrubbed off the LLM.

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.

asciitable.com. Ascii table.

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.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

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.

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.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101.

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.