Machine Unlearning of Pre-trained Large Language Models

Anonymous ACL submission

Abstract

This study investigates the concept of the *'right to be forgotten'* within the context of large language models (LLMs). We explore machine unlearning as a pivotal solution, with a focus on pre-trained models–a notably under-researched area. Our research delineates a comprehensive framework for machine unlearning in pre-trained LLMs, encompassing a critical analysis of seven diverse unlearning methods. We rigorously evaluate these methods against curated datasets sourced from arXiv, books, and GitHub codes, providing a robust benchmark for unlearning performance. Our results show that integrating gradient ascent with gradient descent on in-distribution data improves hyperparameter robustness. We also provide detailed guidelines for efficient hyperparameter tuning in the unlearning process. Our findings advance the discourse on ethical AI practices, offering substantive insights into the mechanics of machine unlearning for pre-trained LLMs and underscoring the potential for responsible AI development.[1]

1 Introduction

Large language models (LLMs) have seen remarkable advancements, attributable to training on extensive and diverse datasets (Ouyang et al., 2022; Wei et al., 2022; Touvron et al., 2023b; Wu et al., 2023; Liang et al., 2023). Yet, the reliance on massive data pools has raised significant ethical concerns, particularly when such data include sensitive, private, or copyrighted material (Li et al., 2022; Shi et al., 2023; Li et al., 2023; Yang et al., 2023).

A prominent example of these issues is the recent lawsuit filed by The New York Times[2] against OpenAI. The lawsuit, responding to the alleged use of millions of articles from The Times in training LLMs like ChatGPT, highlights the critical issue of copyright infringement in LLMs' development.

In response to these ethical challenges, the concept of *machine unlearning* has emerged as a potential remedy. It entails systematically removing specific data from a model's training, ensuring its operation as though the data had never been included (Bourtoule et al., 2021). This approach mitigates the ethical issues stemming from the pre-trained data in LLMs, aligning the technology with evolving legal and ethical standards.

Despite its potential, current research on machine unlearning in the realm of LLMs has been predominantly confined to the fine-tuned model (Kumar et al., 2022; Chen and Yang, 2023; Maini et al., 2024). This focus has limitations, as fine-tuning models on a retained dataset is often feasible, rendering fine-tuning phase unlearning less critical. The real challenge, and our focus in this paper, is the *unlearning of pre-trained LLMs*. This challenge is compounded by several factors: 1) the need to adapt existing unlearning methods from other fields to pre-trained LLMs, 2) the general lack of public availability of pre-trained data used to develop LLMs, and 3) the absence of directly comparable baselines due to the exorbitant costs of retraining pre-trained LLMs.

Our paper addresses them through several key contributions. We first define the problem of machine unlearning for pre-trained LLMs and propose a unified formulation consolidating prior arts under a single unlearning objective. We then investigate seven different unlearning methods in the context of LLMs. To benchmark the unlearning performance, we compile three datasets from sources, including arXiv, books, and GitHub code. Recognizing the impracticality of retraining pre-trained models, we propose an approximate retraining method using an in-distribution, unseen dataset to simulate the performance of a retraining baseline.

Besides, previous studies on machine unlearning

[1] Our code is available at https://anonymous.4open.science/r/Unlearning_LLM-503F

[2] https://nytco-assets.nytimes.com/2023/12/NYT_Complaint_Dec2023.pdf

in LLMs have been limited to small-scale experiments, unlearning at most 128 samples, single corpus sources, or short context lengths (Jang et al., 2023; Eldan and Russinovich, 2023; Yao et al., 2023). In contrast, our work unlearns thousands of chunks, each 4096 tokens in length, from a diverse range of sources across three domains, presenting a more realistic and challenging scenario. Our main contributions and findings are:

- We structure a unified unlearning framework for LLMs, from which seven unlearning methodologies are derived and adapted to LLMs.
- We introduce an approximate retraining evaluation baseline to bypass the impracticality of retraining LLMs. Experiments on three domains demonstrate the efficacy of our methods.
- Gradient ascent combined with gradient descent on in-distribution data shows greater hyperparameter robustness. We offer guidelines to efficiently fine-tune hyperparameters for other methods to streamline and make unlearning more feasible.

We aim to offer a comprehensive solution to unlearning in pre-trained LLMs, contributing to developing more ethical and responsible AI systems.

2 Problem Formulations

Let $\mathcal{D} = \{x_i\}_{i=1}^{N}$ be a training corpus containing N sequences, where $x_{i\in[N]}$ is a sequence of t_i tokens $w_1^i, w_2^i, ..., w_{t_i}^i$. With a slight abuse of notation, we use M to denote both the model itself and its weights. This work focuses on generative LLMs M that are typically trained using the next-token prediction, characterizing the conditional probability given prompts: $P_M(w_{t+1}|w_1, w_2, ..., w_t)$. We denote A as a randomly initialized training algorithm $M \leftarrow A(\mathcal{D})$, where the training objective is to minimize the negative log-likelihood:

$$\mathcal{L}(P_M; \mathcal{D}) = -\sum_{x_i \in \mathcal{D}} \sum_{t=1}^{t_i} \log P_M(w_{t+1}^i | w_1^i, ..., w_t^i).$$

We call the model designated for unlearning the *vanilla* model. We denote a forget set of sequences to be unlearned as $\mathcal{U} \subset \mathcal{D}$. To remove the effect of $\mathcal{U}$, we consider an unlearning algorithm $\widehat{A}$ that applies to $\mathcal{A}(\mathcal{D})$ and outputs an unlearned model M'.

Motivated by differential privacy (Dwork et al., 2006), Ginart et al. (2019) formulated the probabilistic notion of unlearning using (ϵ, δ)-closeness of distributions (Guo et al., 2020; Sekhari et al., 2021). Informally, it requires that the output distributions of $\widehat{A}$ and A run over $(\mathcal{D} \setminus \mathcal{U})$ to be similar. We call it *exact* unlearning if the distributions are identical (*i.e.*, $\epsilon = \delta = 0$); *approximate* unlearning, otherwise. A naïve solution for exact unlearning is just retraining on $\mathcal{D} \setminus \mathcal{U}$ for each $\mathcal{U}$ from scratch. However, it is prohibitively expensive for LLMs, incurring gigantic computation costs and carbon footprints (Luccioni et al., 2022; Zhang et al., 2023a).

Deriving theoretical guarantees for LLMs is also non-trivial, as the underlying transformer architecture is not convex or Lipschitz (Kim et al., 2021). Pragmatically, an active line of research (Golatkar et al., 2020; Chen and Yang, 2023; Kurmanji et al., 2023; Jia et al., 2023) only requires the empirical performance (*e.g.*, classification accuracy) of retrained and unlearned models to be similar. In our context, we can resort to *perplexity* and ensure

$$\mathbb{E}P_{M^\star} \approx \mathbb{E}P_{M'} \text{ with } M^\star \leftarrow A(\mathcal{D} \setminus \mathcal{U}),$$

where $M^\star$ represents the model trained on $\mathcal{D} \setminus \mathcal{U}$. Besides, we require that the validation performance of M' on $\mathcal{D} \setminus \mathcal{U}$ and $\mathcal{U}$ is similar to that of the vanilla model on $\mathcal{D} \setminus \mathcal{U}$ and unseen data, respectively.

3 Unlearning Methods

3.1 Overview

As discussed in Section 2, the objective of LLM unlearning is to *ensure that the model effectively forgets designated token sequences while still preserving its performance on the retain set*. To achieve this goal, we propose an approximate unlearning framework for LLMs using next-token prediction. To unlearn sequences in $\mathcal{U}$, we update the current model M using the gradient derived from

$$\sum_{w \in \mathcal{U}} \sum_{t=1}^{T} \mathbb{E}_{q_t \sim Q_{w_t}} \log P_M(q_t | w_1, w_2, ..., w_{t-1}) + \sum_{z \in \mathcal{R}} \sum_{t=1}^{T} \log P_M(z_t | z_1, z_2, ..., z_{t-1}), \quad (1)$$

where $\mathcal{R} \subseteq \mathcal{D} \setminus \mathcal{U}$ and Q_{w_t} is a set of distributions over token universe $\mathcal{W}$ depending on w_t, which we call *reference distributions*. While Eq. (1) appears complicated, we demonstrate that several iconic unlearning methods are its instances in Section 3.2.

Most existing unlearning approaches (Golatkar et al., 2020; Chen and Yang, 2023; Kurmanji et al.,

2023; Jia et al., 2023) target at (image) classification scenarios. Nevertheless, some of them can be adapted to unlearning token sequences used to train (generative) LLMs with slight modifications.

Below, we focus on those *first-order approximate* unlearning methods that only exploit gradient information and are often more efficient than exact unlearning and second-order designs [3]. Their general formulation is given in Eq. (1), which can be extended for new unlearning methods. We will discuss how to specialize it to each method revisited below and the corresponding pros and cons.

Notably, some methods either consider forgetting an entire class (Tarun et al., 2021) or need to store all intermediate model/gradient information during training (Bourtoule et al., 2021). However, the former is not directly applicable to LLMs, and the latter is too expensive regarding memory. We exclude them from our discussion.

3.2 Approximate Unlearning Methods

3.2.1 Gradient Ascent (or Negative Gradient)

Derived from the general framework outlined in Eq. (1), if we ignore the second term, set $Q_{w_t} = \delta_{w_t}$, and multiply -1 to the gradient, we arrive at the unlearning strategy known as gradient ascent or negative gradient (Golatkar et al., 2020; Jia et al., 2023; Jang et al., 2023). Here, δ_{w_t} is the delta function at w_t such that $q_t \sim Q_{w_t}$ means $q_t = w_t$ with probability 1. The intuition is that M has been trained with $\mathcal{U}$, while the retrained model $M_r^{\mathcal{U}}$ never sees $\mathcal{U}$. Thus, the loss of M on $\mathcal{U}$ is lower than that of $M_r^{\mathcal{U}}$, but the loss should be similar if $|\mathcal{U}|$ is limited. Unfortunately, it is known from the literature that if we perform gradient ascent for too many epochs, the model M will also potentially forget the information about $\mathcal{D} \setminus \mathcal{U}$, thus leading to poor utility. In practice, researchers often only apply gradient ascent in a few epochs.

3.2.2 Fine-tuning with Random Labels

Alternatively, if we ignore the second term of Eq. (1), and set Q_{w_t} to be a uniform distribution over all possible token sets $\mathcal{W}$, we arrive at the strategy known as fine-tuning with random labels, as proposed by Golatkar et al. (2020) for classification problems. The intuition for this strategy is that a model not seeing $\mathcal{U}$ should act as random guessing. While it may seem reasonable at first glance, we argue that uniform distribution for Q_{w_t} is not universally appropriate. For instance, consider the case of two duplicated sequences: one to be unlearned and the other to be retained. Apparently, the retrained model should not act as random guessing on this sequence. In practice, convergence on random labels often leads to a marked decrease in both utility and performance, limiting this method to a brief period of weight adjustment, akin to the earlier mentioned gradient ascent method.

Chundawat et al. (2023); Zhang et al. (2023b) propose to set $Q_{w_t} = P_{M_{\text{rand}}}(w_t|w_1, ..., w_{t-1})$, where M_{rand} is a randomly initialized model (known as incompetent teacher). Their intuition is similar to fine-tuning with random labels, where M_{rand} does not contain information about $\mathcal{U}$. Essentially, these two methods are equivalent, but fine-tuning with random labels is more direct and efficient, so we only adapt it to LLMs.

3.2.3 Unlearning with Adversarial Samples

This approach is originally proposed for classification tasks (Cha et al., 2023). We adapt it to our context below. For simplicity, let us assume only one sequence $w_1, \ldots, w_T$ to be unlearned. We generate adversarial samples $\{a_t\}$ for each t such that they are close to w_t but can confuse M the most

$$a_t = \arg\max_{a \neq w_t} P_M(a|w_1, w_2, \ldots, w_{t-1}). \quad (2)$$

Originally, Cha et al. (2023) proposes choosing adversarial samples a within a small radius to w_t in some metric space for classification tasks. Yet, it is non-trivial to adapt this strategy to LLMs. We thus choose a most likely token a other than w_t.

To unlearn all training sequences in $\mathcal{U}$, we fine-tune M using Eq. (1) while ignoring the second term in Eq. (1) and choosing $Q_{w_t} = \delta_{a_t}$ and $a_t = \arg\max_{a \neq w_t} P_M(a|w_1, w_2, \ldots, w_{t-1})$. The raw approach (Cha et al., 2023) uses $K - 1$ adversarial samples for each training sample to be unlearned in K-class classification problems. Directly using it is not suitable, as generating $|\mathcal{W}| - 1$ adversarial samples for unlearning one token is impractical in our setting. Hence, we simplify it via $Q_{w_t} = \delta_{a_t}$ and $a_t = \arg\max_{a \neq w_t} P_M(a|w_1, w_2, \ldots, w_{t-1})$. Other choices in the same spirit, such as replacing $\arg\max_{a \neq w_t}$ with top-k$_{a \neq w_t}$, are also feasible.

3.2.4 Gradient Ascent + Descent or KL Divergence on Retained Set

On the other hand, disregarding the first term in Eq.(1) leads to a strategy known as fine-tuning on

[3] We present a detailed review on exact unlearning and second-order unlearning methods in Appendix A

the retained set. By updating M with this strategy on $\mathcal{D} \setminus \mathcal{U}$ until convergence, we can achieve the effect of retraining from scratch. For efficiency, researchers typically fine-tune M on a small subset $\mathcal{R} \subseteq \mathcal{D} \setminus \mathcal{U}$ over a few epochs. However, this becomes impractical for LLMs due to the large volume of pre-training data relative to the data designated for unlearning. While not utilized independently here, this method is integrated with gradient ascent techniques, forming a hybrid approach that optimizes both terms in Eq.(1) to balance unlearning effectiveness with utility. To optimize the second term, we adopt both direct gradient ascent and KL-divergence constraint methods, outlined in the prior work (Yao et al., 2023). Moreover, we assess the impact of different data types for the second term, including general pre-training data and domain-specific data matching the unlearning set, termed in-distribution data.

4 Experiments

4.1 Background

Here, we select removing copyrighted data from the pre-trained model as a representative scenario. LLMs have the potential to internalize and reproduce copyrighted content unintentionally. It poses legal challenges and ethical dilemmas, especially when the model's outputs mimic or rephrase the protected material. When it comes to light that copyrighted data has been assimilated into an LLM's training set, machine unlearning techniques can be mobilized to facilitate the model's "forgetfulness" regarding this specific content. Hence, the model's subsequent outputs are safeguarded against undue influences of the copyrighted material, fostering a more compliant and ethical use of data.

4.2 Evaluation Metrics

We focus on evaluating the unlearned models from 1) **Performance on the Forget Set**: The model should not be able to predict correctly on the forget set, or its performance should degrade to the same level as the test set.
2) **Performance on the Retain Set**: Ideally, the model's performance on this set should not degrade significantly, indicating that the unlearning process did not adversely affect the data it should remember. The performance assessment is conducted by measuring the model's accuracy and perplexity on both the forget and retain sets.
3) **Performance on General Downstream Tasks:** We can evaluate the performance on some general downstream tasks, which can provide insights into the model's overall capability post-unlearning. The model's performance on these tasks is expected not to downgrade too much compared with the model before unlearning. The downstream tasks considered include Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2021), the ARC Challenge (Clark et al., 2018), HumanEval (Chen et al., 2021), and Grade School Math (GSM8K) (Cobbe et al., 2021).

Approximate Retraining. To attain ideal unlearning outcomes, one can retrain from scratch to exclude the specified sequences $\mathcal{U}$. This exact unlearning approach is considered as the "gold standard" for evaluating approximate unlearning efficacy, as highlighted in the prior work (Liu et al., 2023). However, retraining LLMs on the entire retained dataset $\mathcal{D}\setminus\mathcal{U}$ is impractical due to substantial computational resource requirements.

To circumvent this challenge, we introduce a surrogate evaluation approach called *approximate retraining*, which is inspired by membership inference attacks (Shokri et al., 2017; Carlini et al., 2022) that identify performance gaps between the training and unseen data. We thus hypothesize that the retrained model will exhibit consistent performance on unseen domain-specific data, albeit inferior to its performance on trained data. Given the significant imbalance between pre-training and unlearning data volumes, we expect the retrained model's performance on unlearned data distributions to closely align with the original (vanilla) model's performance. Consequently, by collecting new data from the same domain as the forget set to create an 'approximate set,' we estimate the retrained model's performance on the forget set by the vanilla model's performance on this approximate set. This estimation can further guide the extent of approximate unlearning, including factors such as the learning rate or optimization steps.

Membership Inference Attack. The complexity of LLMs precludes straightforward interpretable verification of the complete exclusion of specific sequences from the vanilla model. To address this, we employ Membership Inference Attack (MIA) to ascertain whether particular sequences are erased from the LLM's training dataset. This evaluation employs the Min-K% Prob method (Shi et al., 2023), which operates on the premise that

Models	Forget Set			Retain Set		Downstream Task Accuracy ↑				
	ACC↓	PPL↑	MIA↓	ACC↑	PPL↓	MMLU	ARC	HumanEval	GSM8K	Avg.
Vanilla Model	69.02	3.65	50.77	52.68	9.24	63.37	68.49	16.46	33.59	45.48
Approximate Retrain	68.98	3.69	-	-	-	-	-	-	-	-
Gradient Ascent	68.79	3.70	50.28	52.66	9.26	63.45	68.77	15.85	34.04	45.53
Fine-tuning with Random Labels	68.92	3.69	50.55	52.67	9.25	63.37	68.38	14.02	32.22	44.50
Unlearning with Adversarial Samples	68.87	3.69	50.52	52.68	9.25	63.32	68.74	15.24	33.13	45.11
Gradient Ascent + Descent on retain set										
- Descent on in-distribution data	68.87	3.69	50.18	52.66	9.26	63.32	68.52	15.24	33.74	45.21
- Descent on general data	68.81	3.69	50.33	52.93	9.04	63.40	67.87	15.24	33.13	44.91
Gradient Ascent + KL divergence										
- KL on in-distribution data	68.82	3.69	50.29	52.65	9.27	63.40	68.57	15.24	33.89	45.28
- KL on general data	68.79	3.70	50.25	52.65	9.27	63.27	68.38	15.85	33.81	45.33

Table 1: Overall results of unlearning Yi-6B on a subset of pre-training data (500 **arXiv papers**)

Models	Forget Set			Retain Set		Downstream Task Accuracy ↑				
	ACC↓	PPL↑	MIA↓	ACC↑	PPL↓	MMLU	ARC	HumanEval	GSM8K	Avg.
Vanilla Model	80.65	2.40	81.93	52.68	9.24	63.37	68.49	16.46	33.59	45.48
Approximate Retrain	72.91	3.42	-	-	-	-	-	-	-	-
Gradient Ascent	78.19	3.53	74.28	52.60	9.31	63.45	68.40	14.63	35.10	45.40
Fine-tuning with Random Labels	78.00	3.12	80.55	52.50	9.47	62.45	67.02	10.98	29.49	42.48
Unlearning with Adversarial Samples	75.09	3.40	79.51	52.54	9.41	62.36	67.33	9.76	31.39	42.71
Gradient Ascent + Descent on retain set										
- Descent on in-distribution data	76.88	3.45	76.75	52.48	9.38	62.31	66.77	2.44	31.01	40.63
- Descent on general data	78.79	3.57	75.61	53.03	9.00	63.15	67.62	14.63	33.51	44.73
Gradient Ascent + KL divergence										
- KL on in-distribution data	78.78	3.51	76.19	52.61	9.31	63.40	68.21	14.63	34.95	45.30
- KL on general data	78.68	3.58	75.42	52.60	9.31	63.32	68.07	14.02	34.72	45.03

Table 2: Overall results of unlearning Yi-6B on a subset of pre-training data (2K **GitHub code repository files**)

Models	Forget Set			Retain Set		Downstream Task Accuracy ↑				
	ACC↓	PPL↑	MIA↓	ACC↑	PPL↓	MMLU	ARC	HumanEval	GSM8K	Avg.
Vanilla Model	55.26	7.62	74.03	52.68	9.24	63.37	68.49	16.46	33.59	45.48
Approximate Retrain	50.65	10.11	-	-	-	-	-	-	-	-
Gradient Ascent	52.47	9.64	58.47	52.45	9.40	63.32	68.66	16.46	32.90	44.91
Fine-tuning with Random Labels	51.9	10.19	63.69	52.56	9.39	63.05	68.01	16.46	29.64	44.29
Unlearning with Adversarial Samples	52.07	10.02	63.60	52.59	9.35	63.08	68.18	16.46	31.39	44.78
Gradient Ascent + Descent on retain set										
- Descent on in-distribution data	50.07	10.27	56.39	52.34	9.41	63.08	67.70	17.68	29.80	44.57
- Descent on general data	52.49	10.35	69.81	52.88	9.06	63.33	67.78	16.46	32.83	45.10
Gradient Ascent + KL divergence										
- KL on in-distribution data	52.42	10.02	64.02	52.52	9.35	63.50	68.80	16.46	33.59	45.59
- KL on general data	52.85	9.71	62.61	52.58	9.31	63.32	68.55	15.24	32.98	45.02

Table 3: Overall results of unlearning Yi-6B on a subset of pre-training data (100 **Books**)

non-member examples are more prone to containing outlier words with notably high negative log-likelihood values, in contrast to member examples. An important variable affecting the efficacy of MIA is the percentage of tokens with minimal prediction probability; thus, we conduct experiments across various percentages, selecting the one yielding the highest detection performance for each model. The sequence length is set to be 4096 tokens for both member (chunked from the forget set) and non-member (equivalent number of chunks from the approximate set) datasets. The effectiveness of MIA was quantitatively assessed using the Area Under Curve (AUC) metric. Notably, a higher AUC indicates that the targeted sequence is still identifiable within the training set, whereas a score approaching 0.5, indicative of random guess result, suggests superior unlearning effectiveness.

Domains	Forget		Approximate	
	Docs	Chunks	Docs	Chunks
arXiv	500	1,938	6,155	32,144
GitHub	2,000	2,730	15,815	18,929
Books	100	3,038	50	923

Table 4: Document and chunk counts across domains

4.3 Model and Datasets

We conduct experiments using the open-sourced Yi-6B[4] LLM. To rigorously assess the effectiveness of unlearning methods, we perform tests in three distinct settings: arXiv papers, GitHub code repositories, and books. Despite their public availability, these sources may still entail copyright concerns. For instance, although arXiv provides open access to preprints, in many cases, the copyright of each individual preprint remains with its authors or the rights holders. Imagine a case where the paper's authors would like to erase their preprints from the pre-trained LLMs ("the right to be forgotten").

The *forget set* is randomly sampled from the Yi-6B's pre-training data [5], encompassing domains such as arXiv papers, GitHub code repositories, and books. Due to the impracticality of evaluating the model's performance across the entirety of the retained set, we randomly select a sample of 1k sequences from the retained set to create a *general set*. For arXiv papers, the approximate data comprises 6.1k publications from August 2023. The GitHub code repositories' approximate data are 15.8k files from GitHub repositories uploaded in November 2023 with permissive licenses (RealTimeData, 2024). The approximate data of Books are 50 books published after 2023, which are from the unseen data of BookMIA (Shi et al., 2023). We employ the model's maximum input sequence length of 4096 as the chunk length, segmenting the sequences into multiple chunks. All the approximate dataset is preprocessed in the same manner as TogetherComputer (2023), an open-source pre-training data collection to reproduce Llama. The dataset statistics are shown in Table 4.

4.4 Results

We report and analyze the results for unlearning arXiv papers, GitHub code repositories, and books. Given that approximate retraining serves as an optimal target for unlearning, we adjust the learning rate of each experiment to align the results with those achieved through approximate retraining. The number of unlearning epochs is 1. All the experiments are conducted using 8 A800 GPUs.

Unlearning academic papers from arXiv. We task the pre-trained Yi-6B to unlearn 500 academic papers randomly selected from its training data within the arXiv domain. This procedure simulates scenarios in which authors wish to safeguard their proprietary knowledge or unique writing styles. The unlearning performance is shown in Table 1.

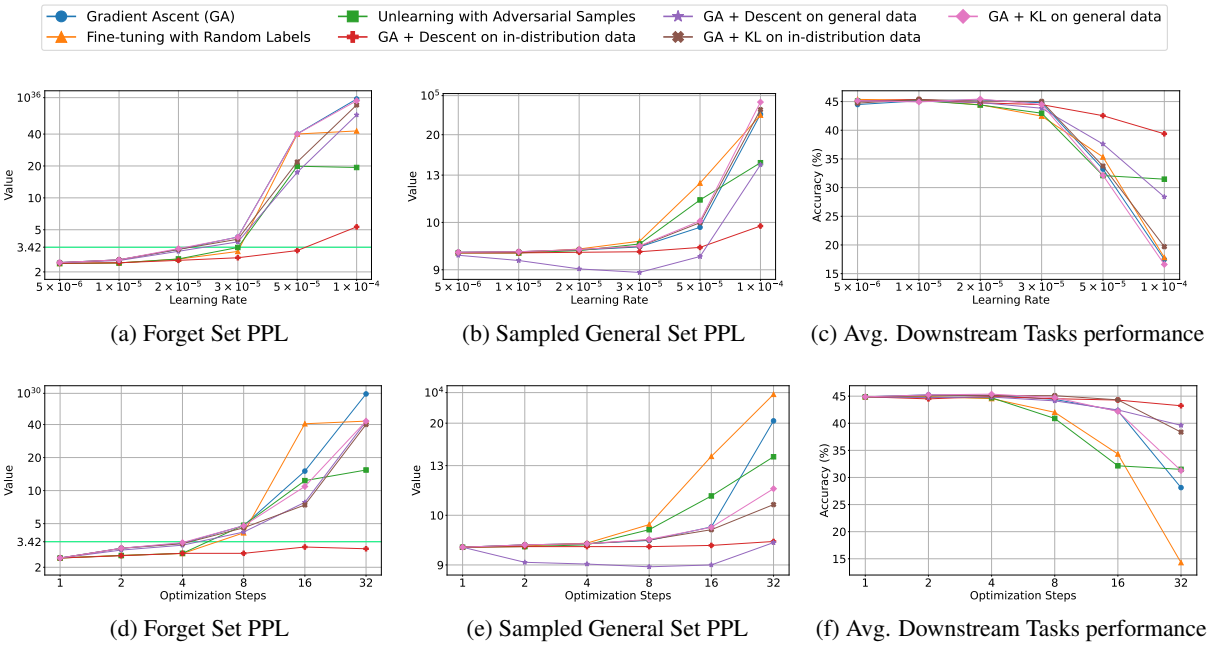
The vanilla model exhibits close performance on both the forget set and unseen approximate data, with perplexity values of 3.65 and 3.69, respectively, demonstrating the model has good generalization capabilities within the arXiv domain. Compared with the vanilla model, unlearned models exhibit a slight decline in next token prediction accuracy (*e.g.*, from 69.02% to 68.79% after gradient ascent), signifying increased difficulty in token extraction given preceding tokens as prompts. The decrease in the AUC score of MIA on unlearned models indicates that it becomes more challenging to differentiate between the forget set and unseen data, suggesting the efficacy of unlearning methods. Notably, only the model subjected to unlearning through a combination of gradient ascent and gradient descent exhibited reduced perplexity on the retain set. This outcome can be attributed to the model learning on a general dataset sampled from the same distribution as the retain set while unlearning the forget set. Furthermore, gradient ascent emerges as the sole method to enhance average accuracy across downstream tasks post-unlearning, underscoring its superiority in maintaining the model's overall utility.

Unlearning programming code from GitHub repositories. We request the pre-trained Yi-6B to unlearn 2000 GitHub code files randomly selected from its training data. This simulates scenarios where developers or organizations seek to remove specific coding patterns, algorithms, or proprietary code from the model's knowledge base, ensuring that their intellectual property remains protected. The unlearning results are displayed in Table 2.

The vanilla model exhibits a more pronounced disparity in performance between the forget set and the unseen approximate set for GitHub code,

[4] https://huggingface.co/01-ai/Yi-6B

[5] We contacted companies with open-sourced LLMs for pre-training data access. Only the Yi model's developers responded, granting us sampled data and permission for its use and open-sourcing.



(a) Forget Set PPL (b) Sampled General Set PPL (c) Avg. Downstream Tasks performance

(d) Forget Set PPL (e) Sampled General Set PPL (f) Avg. Downstream Tasks performance

Figure 1: Figures 1a to 1c are visualization of unlearning results on GitHub code across varying learning rates while optimization steps are fixed at 4. Figures 1d to 1f are visualization of unlearning results on GitHub code across varying optimization steps while learning rate is fixed at 2×10^{-5}. In Figures 1a and 1d, values below 40 are presented on a $\log_{10}$ scale, whereas values above 40 adopt a $\log_{10^{100}}$ scale. The horizontal spring-green line in Figures 1a and 1d delineates the approximate retraining baseline as the unlearning target. For Figures 1b and 1e, the scale transitions from $\log_{10}$ for values under 20 to $\log_{10^{10}}$ for values exceeding 20.

indicating inferior generalization capabilities compared to the arXiv domain. A notable decline in the HumanEval pass@1 score is observed for models unlearned through fine-tuning with random labels, unlearning with adversarial samples, and gradient ascent combined with gradient descent on in-distribution data. Given that HumanEval is a metric specifically designed to assess the code-generating proficiency of LLMs (Chen et al., 2021), this substantial decrease underscores the detrimental impact of these three methods on the model's task-specific utility. Furthermore, a performance reduction of at least 1.83% on the HumanEval pass@1 score is recorded for other methods, suggesting that unlearning GitHub codes from the LLM's pre-training dataset while maintaining coding capabilities presents a challenging task.

Unlearning copyrighted books. We instruct the pre-trained Yi-6B to unlearn 500 books randomly selected from its training data. This process simulates scenarios in which authors or publishers aim to withdraw their literary works, thereby protecting the uniqueness of their content or preventing the model from generating derivative works. The unlearning results are detailed in Table 3.

Similar to the GitHub code domain, the vanilla model's perplexities of 7.62 for the forget set and 10.11 for the approximate set suggest weaker generalization compared to the arXiv domain. Moreover, when fine-tuning to reach perplexity levels similar to approximate retraining, models unlearned via fine-tuning with random labels, adversarial sample unlearning, and the combination of gradient ascent with gradient descent on in-distribution data result in lower accuracy on the forget set than other methods. This indicates that these strategies more effectively obfuscate the model to predict accurately for the forget set, thus better protecting against potential information extraction from the model.

4.5 Ablation studies

Taking the task of unlearning GitHub code as a case study, a series of experiments are conducted to investigate the influence of learning rate and optimization steps on the unlearning outcomes, with results shown in Figure 1. For Figures 1a to 1c, we keep optimization steps fixed at 4 and vary the learning rate between 5×10^{-6} and 1×10^{-4}. In Figures 1d to 1f, the learning rate is constant at 2×10^{-5}, with optimization steps ranging from 1 to 32. We evaluated the unlearned model's perplexity

on both the forget and general sets, along with average performance on downstream tasks, presenting results across seven different unlearning methods for each hyperparameter configuration.

Figure 1 shows that the method combining gradient ascent and descent on in-distribution data is notably tolerant to changes in learning rate and number of optimization steps, indicating high stability. In contrast, other methods exhibit a marked increase in perplexity for both the forget and general sets with higher learning rates or more optimization steps, underscoring their sensitivity to hyperparameter adjustments.

Moreover, since approximate unlearning may either be non-convergent or compromise utility until convergence, it is crucial to conduct a thorough search for the appropriate hyperparameters to ensure optimal unlearning performance. However, the vast search space and lack of definitive reference targets render this task impractical. To address these challenges, we analyze and summarize the following detailed *guidelines* to streamline the hyperparameter adjustment process:

Figure 1d indicates that a high number of optimization steps reduces the stability of the unlearning process, whereas too few steps can average out detailed information over large batches, thereby degrading unlearning quality. Based on these observations, we set the optimization step size to four for optimal balance.

Figure 1a shows that the unlearned model's perplexity on the forget set rises with the learning rate. To find the optimal learning rate aligned with the approximate retraining baseline for unlearning, we recommend starting with a broad granularity search (10^{-5}) within 5×10^{-6} to 5×10^{-5}. This step narrows the search range. A subsequent finer granularity search within this refined interval will identify the learning rate that best achieves the desired unlearning outcomes.

5 Related Work

We provide an overview of current research on machine unlearning, memorization, and forgetting. A more detailed version is deferred to Appendix A.

Machine Unlearning. The concept of machine unlearning is first introduced in Cao and Yang (2015). Bourtoule et al. (2021) further formalizes *exact* unlearning by introducing a general framework: sharded, isolated, sliced, aggregated (SISA). Exact unlearning requires the unlearned model the same as the retrained model. *Approximate* unlearning, which relaxes the requirement, is also explored by bounding the distance (Chourasia and Shah, 2023) or indistinguishability (Sekhari et al., 2021) between the two model's distributions.

Machine unlearning has been extensively researched within the broader field of machine learning (Xu et al., 2024), yet its exploration in generative language models remains limited. Kumar et al. (2022) propose SISA-FC and SISA-A, two computationally efficient extensions of SISA for classification LMs, *e.g.*, BERT. To unlearn knowledge in generative models, Jang et al. (2023) simply perform gradient ascent on target sequences. Eldan and Russinovich (2023) consider a special case of unlearning the Harry Potter books from Llama2-7b. Yao et al. (2023) applies machine unlearning for harmful responses removing and hallucinations eliminating. However, these studies have been limited to fine-tuned models or a single corpus source. Our work explores unlearning pre-trained LLMs on more diverse datasets.

Memorization and Forgetting. Carlini et al. (2019) first quantifies *unintended* memorization by a metric called exposure, revealing severe privacy issues, *e.g.*, membership inference attacks (Carlini et al., 2022) or verbatim data extraction (Carlini et al., 2021). Contrary to memorization, catastrophic forgetting, where a model loses previously learned knowledge when training on new data, has been studied (Kemker et al., 2018; Shao and Feng, 2022). It is a *passive* phenomenon different from unlearning, which actively forces models to "forget" specific samples.

6 Conclusion

In this paper, we investigate the challenge of removing copyrighted pre-training data from LLMs. We present a unified formulation for unlearning LLMs, from which seven unlearning methodologies are derived. We introduce approximate retraining as an evaluation technique to bypass the impracticality of retraining LLMs from scratch. Our experimental analysis across three pre-training data domains validates the efficacy of the unlearning approaches. Furthermore, we find that combining gradient ascent with gradient descent on in-distribution data enhances hyperparameter robustness. We also offer guidelines to streamline the tuning of hyperparameters essential to the unlearning process.

Limitations

This work primarily focuses on conducting experiments with the Yi-6B model. A significant challenge arises since most LLMs do not open-source their pre-training data, making the collection of forget sets infeasible. We encourage future research to investigate the applicability of unlearning processes to other models, including those of larger sizes such as 13B or 70B, or more complicated architecture such as the mixture of experts. Also, we mainly do experiments on three pre-training data domains. Future research should aim to explore unlearning across other domains, including Wikipedia and HackerNews. Besides, our work concentrates on unlearning copyrighted content from LLMs. Future studies could expand our methodologies to address other challenges, such as unlearning biases or harmful outputs in LLMs. Furthermore, since our methods are non-convergent or may reduce model utility until convergence, the adjustment of hyperparameters becomes crucial for ideal unlearning results. While our guidelines simplify and streamline this process, we hope that future research will develop convergent methods that are less dependent on hyperparameter adjustments.

Ethics Statement

In this work, we focus on unlearning pre-trained generative LLMs. Our goal is to enable LLMs to selectively forget particular training sequences while preserving the model's utility. This approach aims to address ethical concerns, including copyright infringement and privacy breaches. The evaluation datasets are compiled from publicly accessible sources, adhering to the licenses associated with the collected data. We also encourage researchers and developers to use our methods responsibly and ethically.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv:2204.05862.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *S&P*, pages 141–159.

Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *S&P*, pages 463–480.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. 2022. Membership inference attacks from first principles. In *S&P*, pages 1897–1914.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *ICLR*.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security*, pages 267–284.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *USENIX Security*, pages 2633–2650.

Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. 2023. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. abs/2301.11578.

Jiaao Chen and Diyi Yang. 2023. Unlearn what you want to forget: Efficient unlearning for llms. arXiv:2310.20150. EMNLP.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion

Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Eli Chien, Chao Pan, and Olgica Milenkovic. 2023. Efficient model updates for approximate unlearning of graph-structured data. In *ICLR*.

Rishav Chourasia and Neil Shah. 2023. Forget unlearning: Towards true data-deletion in machine learning. In *ICML*, pages 6028–6073.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *NeurIPS*, pages 4299–4307.

Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan S. Kankanhalli. 2023. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *AAAI*, pages 7210–7217.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284.

Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407.

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

Sanjam Garg, Shafi Goldwasser, and Prashant Nalini Vasudevan. 2020. Formalizing data deletion in the context of the right to be forgotten. In *EUROCRYPT*, pages 373–402.

Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. 2019. Making AI forget you: Data deletion in machine learning. In *NeurIPS*, pages 3513–3526.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *CVPR*, pages 9301–9309.

Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. 2020. Certified data removal from machine learning models. In *ICML*, pages 3832–3842.

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. 2021. Adaptive machine unlearning. In *NeurIPS*, pages 16319–16330.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *ICLR*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *ICML*, pages 2790–2799.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*.

Yiyang Huang and Clément L. Canonne. 2023. Tight bounds for machine unlearning via differential privacy. arXiv:2309.00886.

Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. 2021. Approximate data deletion from machine learning models. In *AISTATS*, pages 2008–2016.

Matthew Jagielski, Om Thakkar, Florian Tramèr, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Guha Thakurta, Nicolas Papernot, and Chiyuan Zhang. 2023. Measuring forgetting of memorized training examples. In *ICLR*.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. Knowledge unlearning for mitigating privacy risks in language models. In *ACL*, pages 14389–14408.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. 2023. Model sparsification can simplify machine unlearning. In *NeurIPS (Spotlight)*. ArXiv:2304.04934.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler L. Hayes, and Christopher Kanan. 2018. Measuring catastrophic forgetting in neural networks. In *AAAI*, pages 3390–3398.

Hyunjik Kim, George Papamakarios, and Andriy Mnih. 2021. The lipschitz constant of self-attention. In *ICML*, pages 5562–5571.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver

Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations - democratizing large language model alignment.

Vinayshekhar Bannihatti Kumar, Rashmi Gangadharaiah, and Dan Roth. 2022. Privacy adhering machine un-learning in NLP. arXiv:2212.09573.

Meghdad Kurmanji, Peter Triantafillou, and Eleni Triantafillou. 2023. Towards unbounded machine unlearning. arXiv:2302.09880.

Kecen Li, Chen Gong, Zhixiang Li, Yuzhong Zhao, Xinwen Hou, and Tianhao Wang. 2023. Meticulously selecting 1% of the dataset for pre-training! generating differentially private images data with semantics query. *arXiv preprint arXiv:2311.12850*.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2022. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models.

Zheyuan Liu, Guangyao Dou, Yijun Tian, Chunhui Zhang, Eli Chien, and Ziwei Zhu. 2023. Breaking the trilemma of privacy, utility, efficiency via controllable machine unlearning.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. 2022. Estimating the carbon footprint of bloom, a 176b parameter language model. arXiv:2211.02001.

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

Ilya Mironov. 2017. Rényi differential privacy. In *CSF*, pages 263–275.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Alexandra Peste, Dan Alistarh, and Christoph H. Lampert. 2021. SSSE: efficiently erasing samples from trained machine learning models. arXiv:2107.03860.

RealTimeData. 2024. github_latest.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. 2021. Remember what you want to forget: Algorithms for machine unlearning. In *NeurIPS*, pages 18075–18086.

Chenze Shao and Yang Feng. 2022. Overcoming catastrophic forgetting beyond continual learning: Balanced training for neural machine translation. In *ACL*, pages 2023–2036.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *S&P*, pages 3–18.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *NeurIPS*.

Ayush K. Tarun, Vikram S. Chundawat, Murari Mandal, and Mohan S. Kankanhalli. 2021. Fast yet effective machine unlearning. *CoRR*, abs/2111.08947.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. In *NeurIPS*.

TogetherComputer. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa,

Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288.

Lingzhi Wang, Xingshan Zeng, Jinsong Guo, Kam-Fai Wong, and Georg Gottlob. 2024. Selective forgetting: Advancing machine unlearning techniques and evaluation in language models. *arXiv preprint arXiv:2402.05813*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. 2024. Machine unlearning: A survey. *ACM Comput. Surv.*, 56(1):9:1–9:36.

Borui Yang, Wei Li, Liyao Xiang, and Bo Li. 2023. Towards code watermarking with dual-channel transformations. *arXiv preprint arXiv:2309.00860*.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2023. Large language model unlearning. In *Socially Responsible Language Modelling Research*.

Dawen Zhang, Pamela Finckenberg-Broman, Thong Hoang, Shidong Pan, Zhenchang Xing, Mark Staples, and Xiwei Xu. 2023a. Right to be forgotten in the era of large language models: Implications, challenges, and solutions. arXiv:2307.03941.

Xulong Zhang, Jianzong Wang, Ning Cheng, Yifu Sun, Chuanyao Zhang, and Jing Xiao. 2023b. Machine unlearning methodology base on stochastic teacher network. arXiv:2308.14322.

A Related Work (Full Version)

In this section, we provide greater detail about related work on machine unlearning, memorization and forgetting, the relation between machine unlearning, differential privacy, and alignment, exact unlearning, and second-order methods of machine unlearning.

Machine Unlearning. Cao and Yang (2015) introduce the notion of machine unlearning. They give a heuristic method, transforming learning algorithms into a summation form for forgetting data lineage. Their goal is to ensure that unlearned models exactly match the ones retrained from scratch. Subsequently, it is formalized as *exact* unlearning or data deletion (Ginart et al., 2019; Bourtoule et al., 2021) of a specific training sample, requiring the distributions of unlearned and retrained models are identical. Ginart et al. (2019) propose two tailored approaches for k-means, while Bourtoule et al. (2021) propose a general unlearning framework: sharded, isolated, sliced, aggregated (SISA).

Exact unlearning may be too "strong" to achieve; it can be "relaxed" to *approximate* unlearning (Ginart et al., 2019) by bounding the "distance" (*e.g.*, Rényi divergence (Chourasia and Shah, 2023) or indistinguishability (Sekhari et al., 2021)) between the two models' distributions. More generally, one can unlearn a subset of training points (Sekhari et al., 2021) that can even be adaptively chosen (Gupta et al., 2021).

Since the seminal proposal, machine unlearning has been widely studied in ML in general (Xu et al., 2024) but remains rarely explored in NLP, notably generative LLMs. Zhang et al. (2023a) discusses the challenges and implications of unlearning and other approaches to realize RTBF in LLMs. Kumar et al. (2022) propose SISA-FC and SISA-A, two extensions of SISA for classification LMs, *e.g.*, BERT. SISA-FC only trains fully connected task layers, and SISA-A resorts to Adapters (Houlsby et al., 2019) that only update a few plug-in parameters. Chen and Yang (2023) propose an efficient unlearning method via a selective teacher-student formulation for both classification and summarization tasks. They also design a fusion mechanism to merge unlearning layers for sequential data forgetting. To unlearn knowledge in generative models, Jang et al. (2023) simply perform gradient ascent on target sequences. Wang et al. (2024) presents a selective unlearning method to minimize negative impacts on unlearned model's capabilities and proposes evaluation metrics focusing on sensitive information. Eldan and Russinovich (2023) consider a special case of unlearning the Harry Potter books from Llama2-7b. They first use a reinforced model to identify the tokens that are most related to the unlearning target and then replace idiosyncratic terms with generic ones to generate alternative labels for fine-tuning the model. Yao et al. (2023) applies machine unlearning for harmful responses removing and hallucinations eliminating. Maini et al. (2024) presents a benchmark for unlearning fictitious authors on fine-tuned models.

Memorization and Forgetting. Training data memorization to some extent is pivotal for model generalization, but *unintended* memorization, first quantified by a metric called exposure (Carlini et al., 2019), poses severe privacy issues, *e.g.*, membership inference attacks (Carlini et al., 2022) or verbatim data extraction (Carlini et al., 2021). Carlini et al. (2023) illustrate that memorization relies on the model scale, training data deduplication, and prompting context length.

As opposed to memorization, catastrophic forgetting, which means that a model tends to forget previously learned knowledge when training on new data, has been studied (Kemker et al., 2018; Shao and Feng, 2022). It is a *passive* phenomenon different from unlearning, which actively forces models to "forget" specific samples. As with memorization, concurrent works (Tirumala et al., 2022; Jagielski et al., 2023) define and measure forgetting as a form of privacy leakage.

Machine Unlearning vs. Differential Privacy. DP is a rigorous framework for protecting individual privacy in data analytics by adding calibrated noise to query results (Dwork and Roth, 2014). Definition-wise, approximate unlearning is reminiscent of DP. They use the same metric for distributional closeness (*e.g.*, (ϵ, δ)-indistinguishability (Guo et al., 2020)) but with a substantial difference. Unlearning compares two algorithms–unlearning and retraining–on the same dataset, whereas DP compares the same algorithm run on neighboring datasets (differing in an individual's data). DP is a sufficient (not necessary) condition for unlearning (Guo et al., 2020): An DP mechanism working on datasets with edit distance m naturally unlearns *any* m samples. Prior DP-based unlearning designs (Guo et al., 2020;

Sekhari et al., 2021; Izzo et al., 2021) often assume convex loss functions. Sekhari et al. (2021); Huang and Canonne (2023) bound the "deletion" capacity (*i.e.*, how many samples can be unlearned while ensuring desirable loss) better than m in DP. Also, DP can mitigate the adaptivity of unlearning requests (Gupta et al., 2021). Garg et al. (2020) provide an alternative definitional framework for RTBF from cryptographic primitives.

Machine Unlearning vs. Alignment. Alignment in LLMs, the process of adjusting these models to resonate with human values, is typically accomplished through techniques like supervised fine-tuning (SFT) (Ouyang et al., 2022; Conover et al., 2023; Chiang et al., 2023; Köpf et al., 2023; Touvron et al., 2023b) and reinforcement learning with human feedback (RLHF) (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022; Touvron et al., 2023a,b). These methods rely on human-generated demonstrations or rewards and penalty systems. Machine unlearning, on the other hand, uniquely focuses not on promoting correct behavior but discouraging outputs misaligned with human values (Yao et al., 2023). This method thus offers a complementary approach to standard SFT techniques.

Second-order methods of Machine Unlearning. The high-level idea of almost all second-order unlearning methods is through Taylor expansion on the gradient at the stationary point. This leads to a Newton-type update, which involves Hessian inverse (or its approximation) computation (Golatkar et al., 2020; Peste et al., 2021). Apparently, the hessian-related operation is prohibitive to LLM due to its billions if not trillions parameters. Nevertheless, current theoretical approximate unlearning approaches with privacy guarantees are all second-order to the best of our knowledge (Guo et al., 2020; Sekhari et al., 2021; Chien et al., 2023). We decided to briefly introduce them in the context of classification and discuss the potential way of extending them for LLM.

Here, we denote M for both the LLM and its model parameters with a slight abuse of notation. Assume M is well-trained with respect to the training loss $\mathcal{L}(P_M; \mathcal{D})$ so that it is a stationary point $\nabla\mathcal{L}(P_M; \mathcal{D}) = 0$. Similarly, the retrain model $M_r^{\mathcal{U}}$ is also a stationary point with respect to the loss $\mathcal{L}(P_{M_r^{\mathcal{U}}}; \mathcal{D} \setminus \mathcal{U})$, so that $\nabla\mathcal{L}(P_{M_r^{\mathcal{U}}}; \mathcal{D} \setminus \mathcal{U}) = 0$. We can apply a first order Tayler expansion on $\nabla\mathcal{L}(P_{M_r^{\mathcal{U}}}; \mathcal{D} \setminus \mathcal{U})$ at M, which leads to

$$\nabla\mathcal{L}(P_M; \mathcal{D} \setminus \mathcal{U}) + \nabla^2\mathcal{L}(P_M; \mathcal{D} \setminus \mathcal{U})(M_r^{\mathcal{U}} - M) \approx 0$$
$$\Rightarrow M_r^{\mathcal{U}} \approx M - (\nabla^2\mathcal{L}(P_M; \mathcal{D} \setminus \mathcal{U}))^{-1}\nabla\mathcal{L}(P_M; \mathcal{D} \setminus \mathcal{U}).$$

The theoretical unlearning approach will further introduce some privacy noise (similar to the Gaussian mechanism (Mironov, 2017)) to obfuscate the potential privacy leakage rigorously (Guo et al., 2020; Sekhari et al., 2021; Chien et al., 2023), where the noise variance determined by the worst-case error of the Taylor approximation. This analysis can only be done for strongly convex problems with additional smoothness assumptions and is thus not applicable to LLMs. In practice, some researchers still follow this second-order update with a heuristic-based noise addition design, which has demonstrated superior performance on privacy and utility (Golatkar et al., 2020). The main focus of this direction is to improve the computation complexity of the second-order update, with ideas leveraging the Fisher information matrix (Golatkar et al., 2020) and rank one update by Sherman-Morrison lemma (Peste et al., 2021). Nevertheless, the computation complexity of these advanced second-order methods is still too expensive for LLM. One potential direction is to apply these second-order updates only for adaptor (Houlsby et al., 2019) or LoRA (Hu et al., 2022), which contain much fewer parameters to be modified. It remains open whether it is possible to apply second-order methods for LLM or not at the moment.

Exact Unlearning. We introduce exact unlearning methods that correspond to (ϵ, δ) notion of unlearning with $\epsilon = \delta = 0$. While ensuring $\epsilon = \delta = 0$ is desired in some extreme cases, it generally fails to explore the beneficial trade-off between unlearning quality, model utility, and time complexity.

SISA (Sharded, Isolated, Sliced, and Aggregated) framework (Bourtoule et al., 2021) is a general approach to achieve exact unlearning for general deep neural networks at the cost of changing the training pipeline significantly. The main idea is to partition the training dataset $\mathcal{D}$ into K disjoint sets $\mathcal{D}_1, \ldots, \mathcal{D}_K$. For each $\mathcal{D}_i$, we train or fine-tune M on it independently, which results in K models $M_i = A(\mathcal{D}_i)$. We use any fixed aggregation strategy for these K models for the final prediction or output. Unlearning in the SISA framework is straightforward. Given an unlearning request $\mathcal{U}$, we retrain all models M_i for i such

that $\mathcal{U} \cap \mathcal{D}_i \neq \emptyset$. Apparently, storing K copies of LLMs is memory-expensive and impractical which makes the SISA approach not applicable to the pre-training task. The authors of (Kumar et al., 2022) leverage the SISA approach to the fine-tuning task by only fine-tuning a fully connected layer (FC) or Adapter (A) on top of a freeze public pretrained model M. They termed these methods SISA-FC and SISA-A respectively.

The SISA framework is currently the only method providing a theoretical privacy guarantee while applicable to LLMs. However, the efficiency and utility of this approach are greatly affected by the choice of K and dataset dependent. Clearly, when $K = 1$ we simply arrive at retraining from scratch, which maximally preserves the utility but exhibits impractical time complexity. Choosing a large K can improve the efficiency but may degrade model utility. It is unclear at the moment how to choose an appropriate K.