TOWARDS ROBUST AND COST-EFFICIENT KNOWLEDGE UNLEARNING FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have demonstrated strong reasoning and memorization capabilities via pretraining on massive textual corpora. However, this poses risk of privacy and copyright violations, highlighting the need for efficient machine unlearning methods that remove sensitive data without retraining from scratch. While Gradient Ascent (GA) is commonly used to unlearn by reducing the likelihood of generating unwanted content, it leads to unstable optimization and catastrophic forgetting of retrained knowledge. We also find that combining GA with low-rank adaptation results in poor trade-offs between computational cost and generative performance. To address these challenges, we propose two novel techniques for robust and efficient unlearning for LLMs. First, we introduce Inverted Hinge Loss, which suppresses unwanted tokens while maintaining fluency by boosting the probability of the next most likely token. Second, we develop a data-adaptive initialization for LoRA adapters via low-rank approximation weighted with relative Fisher information, thereby focusing updates on parameters critical for removing targeted knowledge. Experiments on the Training Data Extraction Challenge dataset using GPT-Neo models as well as on the TOFU benchmark with Phi-1.5B and Llama2-7B models demonstrate that our approach effectively removes sensitive information while maintaining reasoning and generative capabilities with minimal impact.

1 INTRODUCTION

Large Language Models (LLMs) exhibit substantial performance gains in downstream tasks with increasing model size and amount of pretraining data (Zhao et al., 2023). This has prompted extensive research on collecting high-quality textual corpora for LLM pretraining and developing larger models to an unprecedented scale (Brown et al., 2020; Chowdhery et al., 2023; Smith et al., 2022; Rae et al., 2021; Dubey et al., 2024). However, this approach has introduced significant privacy concerns due to LLMs' tendency to memorize data indiscriminately (Carlini et al., 2021; 2023). For instance, Personally Identifiable Information (*e.g.*, names, phone numbers, and email addresses) can be easily extracted from LLMs (Carlini et al., 2021). Additionally, OpenAI is facing multiple copyright infringement lawsuits due to unpermitted use of licensed articles during LLM pretraining (Grynbaum & Mac, 2023). In response to such challenges as well as increasing interest in one's right to be forgotten (*e.g.*, the GDPR legislation) (Voigt & Von dem Bussche, 2017; Rosen, 2011; Villaronga et al., 2018), machine unlearning for LLMs has emerged a critical and rapidly growing research field (Yao et al., 2023; Si et al., 2023).

One method for LLM unlearning would be to filter out sensitive data from the corpus and retrain the model from scratch, an approach known as *exact* unlearning. With unprecedentedly large models and pretraining datasets, this process is highly resource-intensive and can easily become intractable under the possibility of multiple data deletion requests made in a sequential manner. This motivates *approximate* unlearning, where the goal is to remove knowledge of specific data instances without retraining the model from scratch (Figure 1). In this regard, several novel approaches have been proposed for knowledge unlearning. For instance, Jang et al. (2023) introduced a simple method that finetunes LLMs using Gradient Ascent (GA) on data requested for deletion and also proposed n-gram-based metrics to evaluate its effectiveness. Meanwhile, knowledge distillation-based methods, such as those proposed by Wang et al. (2023) and Liu et al. (2024), aim to transfer knowledge selectively to a secondary model for unlearning. However, both GA and knowledge distillation-based methods

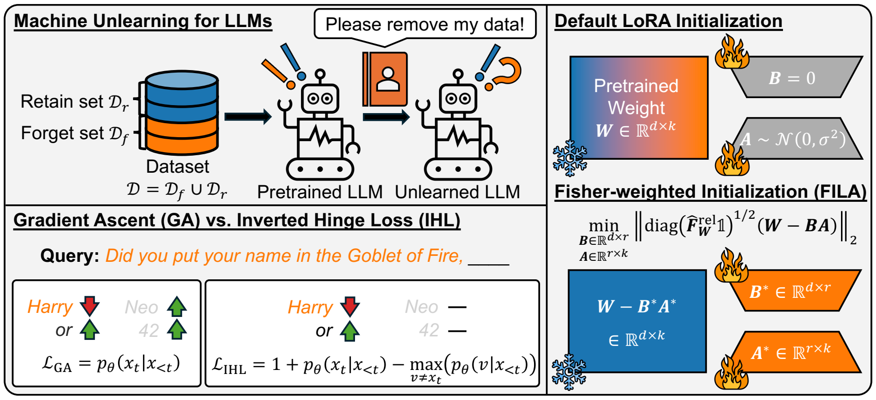

Machine Unlearning for LLMs
Please remove my data!
Retain set $\mathcal{D}_r$
Forget set $\mathcal{D}_f$
Dataset
$\mathcal{D} = \mathcal{D}_f \cup \mathcal{D}_r$
Pretrained LLM
Unlearned LLM
Default LoRA Initialization
Pretrained Weight $W \in \mathbb{R}^{d\times k}$
$B = 0$
$A \sim \mathcal{N}(0, \sigma^2)$
Fisher-weighted Initialization (FILA)
$\min_{B\in\mathbb{R}^{d\times r}, A\in\mathbb{R}^{r\times k}} \left\| \mathrm{diag}(\hat{F}_W^{\mathrm{rel}}\mathbb{1})^{1/2}(W - BA) \right\|_2$
$W - B^*A^* \in \mathbb{R}^{d\times k}$
$B^* \in \mathbb{R}^{d\times r}$
$A^* \in \mathbb{R}^{r\times k}$
Gradient Ascent (GA) vs. Inverted Hinge Loss (IHL)
Query: Did you put your name in the Goblet of Fire, ____
Harry or Neo 42
$\mathcal{L}_{\mathrm{GA}} = p_\theta(x_t|x_{<t})$
Harry or Neo — 42 —
$\mathcal{L}_{\mathrm{IHL}} = 1 + p_\theta(x_t|x_{<t}) - \max_{v\neq x_t}(p_\theta(v|x_{<t}))$

Figure 1: LLM unlearning aims to forget data points in $\mathcal{D}_f$ while maintaining knowledge of the retain set $\mathcal{D}_r$. Unlike GA, our IHL induces higher unlearning stability by reducing the likelihood of unwanted tokens in a controlled manner. To accelerate unlearning with IHL, FILA extracts and places parameters important in generating $\mathcal{D}_f$ to LoRA weights a priori via weighted low-rank approximation. IHL and FILA form a powerful synergy towards robust and efficient LLM unlearning.

face significant challenges: GA suffers from unstable optimization due to the unbounded nature of the objective loss, while distillation-based methods incur substantial computational costs from relying on a secondary model. Above all, these approaches share a critical drawback: the high computational cost of full fine-tuning all parameters within the LLMs.

Meanwhile, Low-Rank Adaptation (LoRA) has emerged as one of the most prominent techniques for parameter-efficient fine-tuning on downstream tasks (Hu et al., 2022). The core idea of LoRA is to freeze all pretrained weights and instead train low-rank decomposition matrices to model the weight changes in each linear layer, effectively reducing the number of trainable parameters and thus its memory cost. In addition to its efficiency, the low-rankness in LoRA also induces a powerful regularization (Biderman et al., 2024), which can be beneficial in the realm of LLM unlearning by stabilizing optimization and preventing catastrophic forgetting of other remaining knowledge. From this perspective, we conjecture that LoRA could be a valuable approach in practical unlearning scenarios. Recent studies have explored using LoRA for LLM unlearning: Chen et al. (2024) applied LoRA for single-step unlearning, while Gao et al. (2024) proposed orthogonal LoRA for continual unlearning. However, these studies focus on applying existing LoRA variants to specific unlearning scenarios, without discussion on designing fast and cost-efficient LoRA for unlearning in LLMs.

In this paper, we explore LLM unlearning under the low-rank adaptation paradigm and propose two novel techniques for efficient knowledge unlearning. First, we analyze the derivatives of GA and highlight its issues: 1) gradients increasing the probability of all tokens cause unnecessary forgetting, and 2) maximizing next-token loss can lead to unbounded optimization and divergence. To address these issues, we propose the Inverted Hinge Loss (IHL) that aims to replace each token to unlearn with the next most-probable token, and show that IHL enables fast and stable tuning by resolving the issues of GA. Second, we find that the low-rank regularization in LoRA is too strong when unlearning with IHL, leading to suboptimal cost vs. post-unlearning performance trade-offs. To address this, we propose Fisher-Initialization of Low-rank Adapters (FILA), which data-adaptively assigns parameters responsible for generating unwanted information to adapters prior to tuning by decomposing the pretrained parameters weighted by the relative Fisher-information matrix (See Figure 1). Experiments on the Training Data Extraction Challenge dataset (GPT-Neo) and the TOFU benchmark (Phi-1.5B, Llama2-7B) demonstrate that IHL combined with FILA outperforms existing baselines in both efficiency and post-unlearning performance. In summary, the main contributions of our study are as follows:

- We analyze the shortcomings of GA—unbounded optimization and unnecessary forgetting—through its derivative and propose IHL to address these issues.
- We introduce FILA, a method to accelerate unlearning by data-adaptively assigning parameters responsible for unwanted information to low-rank adapters.
- We demonstrate that IHL combined with FILA outperforms previous baselines in terms of both efficiency and post-unlearning performance (See Figure 2).

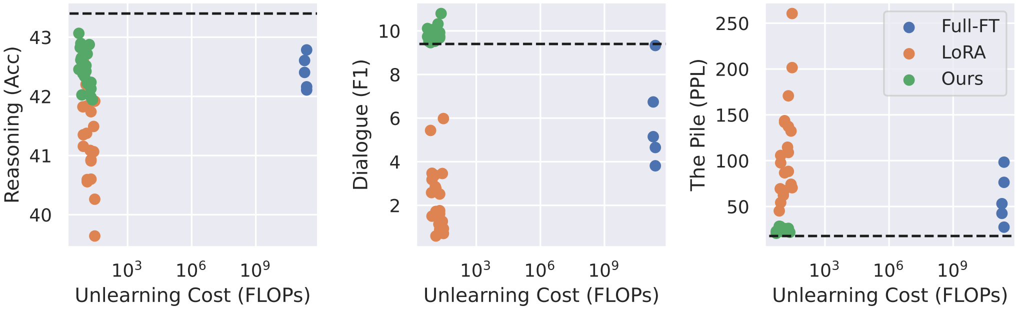


Figure 2: Compute cost for successful unlearning vs. post-unlearning downstream performances. We unlearn 32 randomly sampled sequences from the Training Data Extraction Challenge from GPT-Neo-125M. Each point represents a different forget set and LoRA rank (if used). **Left:** Accuracy averaged across 9 classification tasks (higher is better). **Middle:** F1 score averaged across 4 dialogue generation tasks (higher is better). **Right:** Perplexity on the validation set of the Pile dataset (lower is better). Dashed lines indicate the performances of the model prior to unlearning. Unlearning via gradient differences (GD) with vanilla LoRA leads to significant loss in performance compared to full-parameter GD unlearning due to lack of plasticity. However, our proposed method using both the Inverted Hinge Loss and Fisher-weighted LoRA initialization performs competitively to unlearning via full-finetuning in all three aspects while enjoying the cost-efficiency of LoRA.

2 RELATED WORK

Machine Unlearning. The primary objective of machine unlearning is to adapt a pretrained model to discard information acquired from a specific subset of data previously used during pretraining, with active research focused on image classification (Cao & Yang, 2015; Golatkar et al., 2020; Tarun et al., 2023; Mehta et al., 2022; Chundawat et al., 2023; Cha et al., 2024). Recently, its significance has grown notably with Large Language Models (LLMs) due to crucial need for managing unintended memorization of pretraining data intrinsic to LLMs (Si et al., 2023; Yao et al., 2024b). Several machine unlearning algorithms for LLMs focus on parameter optimization (Si et al., 2023). For example, Wang et al. (2023) introduced Knowledge Gap Alignment, using knowledge distillation between models trained on different datasets. Chen & Yang (2023) proposed an unlearning layer to selectively remove specific knowledge while preserving other parameters, while Liu et al. (2024) developed a two-stage framework to capture and negate harmful knowledge. However, these approaches are limited by the need to retain large datasets (Wang et al., 2023; Chen & Yang, 2023) or rely on secondary models for distillation (Wang et al., 2023; Liu et al., 2024). Model-editing methods, such as task arithmetic for suppressing harmful content (Ilharco et al., 2023; Wu et al., 2023), avoid substantial costs but show limited effectiveness in unlearning. In contrast, Jang et al. (2023) used Gradient Ascent (GA) for LLM unlearning by maximizing the next-token prediction loss on the forget data, effectively unlearning while preserving model performance. Following this, GA has become a standard baseline, with Yao et al. (2024a) improving its robustness by combining GA with gradient descent on in-distribution data.

Parameter-Efficient Fine-Tuning. Fine-tuning large language models (LLMs) for specific tasks is computationally expensive due to their size. To address this, Parameter-Efficient Fine-Tuning (PEFT) methods adapt only a small subset of parameters while keeping the pretrained ones frozen (Liu et al., 2022b; Qiu et al., 2023; Liu et al., 2023). Inspired by the small intrinsic rank of LLMs (Li et al., 2018; Aghajanyan et al., 2021), LoRA and its derivatives add low-rank adapters to the model's linear layers (Hu et al., 2022; Zhang et al., 2023; Yeh et al., 2023; Kopiczko et al., 2024; yang Liu et al., 2024). These adapters can be merged with pretrained parameters after fine-tuning, maintaining the original inference cost. While most methods use random initialization for LoRA adapters, PiSSA (Meng et al., 2024) suggests initializing them using the principal singular vectors and values of the linear weights.

Although various methods in machine unlearning and parameter-efficient finetuning for LLMs have been discussed, this paper focuses on analyzing the inherent issues of GA, introducing a novel unlearning loss function to overcome these issues, and exploring cost-effective unlearning methods that do not require full finetuning. These areas have not been adequately addressed in previous studies, highlighting the contributions of our work.

3 PROPOSED METHOD

3.1 PRELIMINARIES

Problem and notation. Given a sequence of T tokens $\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$, a language model (LM) models the likelihood of the sequence via next-token prediction: $p_\theta(\boldsymbol{x}) = \prod_{t=1}^{T} p_\theta(x_t|x_{<t})$. After pretraining, we assume that an end-user has requested to delete a subset of the training set $\mathcal{D}_f \subset \mathcal{D}$, which we refer to as the *forget set*. The *retain set* $\mathcal{D}_r$ refers to an auxiliary dataset that contains other relevant knowledge that must be retained after unlearning (*e.g.*, Wikitext; Merity et al. 2017).

Gradient Ascent. Ideally, the LM must assign low probability to sequences in $\mathcal{D}_f$, leading to a simple yet effective baseline of Gradient Ascent (GA; Jang et al. 2023). GA unlearns a sequence of tokens $\boldsymbol{x} = (x_1, \ldots, x_T)$ by maximizing the next-token prediction loss:

$$\mathcal{L}_{\text{GA}}(\boldsymbol{x}) = -\sum_{t=1}^{T} \log(p_\theta(x_t|x_{<t})). \tag{1}$$

In practice, the log-likelihood is computed using cross-entropy loss and thus we GA essentially minimizes the Negative Cross-Entropy (NCE) loss. Therefore, GA maximizing the next-token prediction loss involves unbounded optimization, leading to an ill-posed process with unstable tuning. While Gradient Difference (GD) aims to alleviate this instability by minimizing the next-token prediction loss for $\mathcal{D}_r$ alongside NCE on $\mathcal{D}_f$ as regularization, we find that the approach falls short of a fundamental solution, showing performance degradation as unlearning updates are made.

Low-Rank Adaptation. Based on the assumption that parameter changes due to LLM adaptation exhibits an intrinsic low-rank (Aghajanyan et al., 2021), LoRA models the change in parameters $\Delta\boldsymbol{W} \in \mathbb{R}^{d\times k}$ of each linear weight $\boldsymbol{W} \in \mathbb{R}^{d\times k}$ via a product of two low-rank matrices $\boldsymbol{A} \in \mathbb{R}^{r\times k}$ and $\boldsymbol{B} \in \mathbb{R}^{d\times r}$ where $r \ll \min(d, k)$ is the rank of the LoRA adapter. In other words, the output of the adapted linear layer given an input $\boldsymbol{x}$ becomes:

$$(\boldsymbol{W} + \Delta\boldsymbol{W})\boldsymbol{x} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{A}\boldsymbol{x}.$$

During fine-tuning, the original weight $\boldsymbol{W}$ is kept frozen and only the low-rank factors $\boldsymbol{A}$ and $\boldsymbol{B}$ are updated via gradient descent. To ensure that the initial attachment of LoRA adapters does not alter the output of the LLM, LoRA defaults to initializing $\boldsymbol{A}$ with a Kaiming-uniform distribution (He et al., 2015) and $\boldsymbol{B}$ as the zero matrix. After finetuning, LoRA adapters can simply be merged with the original weights $\boldsymbol{W}' = \boldsymbol{W} + \boldsymbol{B}\boldsymbol{A}$, thereby avoiding any additional latency during inference.

3.2 PRELIMINARY RESULTS

Despite its wide use in domain adaptation and instruction tuning, LoRA is not yet explored under the task of LLM unlearning to the best of our knowledge. Therefore, we first share empirical results from low-rank adapting LLMs using GD as our objective to motivate our approach. Figure 2 shows the results. Notably, vanilla LoRA suffers from lack of plasticity and ends up failing to sufficiently unlearn $\mathcal{D}_f$ within 20 epochs. When running more unlearning epochs or increasing the learning rate for sufficient unlearning, the model loses its previously acquired reasoning and generative capabilities, as shown in the significant decrease in Reasoning and Dialogue performances. In the remainder of this section, we present two techniques towards making LLM unlearning viable while enjoying the efficiency of LoRA.

3.3 INVERTED HINGE LOSS: A NOVEL LOSS FUNCTION FOR LLM UNLEARNING

Motivation. We analyze the inherent issues of GA from the perspective of its derivative. The output layer of a language model is a softmax layer that outputs probabilities over the vocabulary. Let y_t be the logits (pre-softmax activations) produced by the LLM model for the t-th token, and let V be the vocabulary size. The probability $p_\theta(x_t|x_{<t})$ is given by the softmax function: $p_\theta(x_t|x_{<t}) = \exp(y_t^{(x_t)})/\sum_{v=1}^{V}\exp(y_t^{(v)})$ where $y_t^{(x_t)}$is the logit corresponding to the true token x_t and $y_t^{(v)}$ is the logit corresponding to the v-th token in the vocabulary. When we use $\mathcal{L}_{\text{GA}}$ for unlearning for LLMs, the gradient of the log-probability with respect to the logits is

$$\frac{\partial \log\left(p_\theta(x_t|x_{<t})\right)}{\partial y_t^{(v)}} = \begin{cases} 1 - p_\theta(x_t|x_{<t}) & \text{if } v = x_t \\ -p_\theta(v|x_{<t}) & \text{if } v \neq x_t \end{cases}$$

From this derivative of GA, we can interpret its unlearning mechanism: given the prefix $x_{<t}$, GA reduces the prediction score of the true token x_t in proportion to $1 - p_\theta(x_t|x_{<t})$ while increasing the scores of other tokens (*i.e.*, $v \neq x_t$) by $p_\theta(v|x_{<t})$. This process effectively shifts the model's prediction for $x_{<t}$ away from the true token x_t, thereby achieving unlearning. However, we can confirm that GA suffers form the following problems during unlearning: (1) **Gradient spread**, where reducing the score of x_t while increasing the scores of all other tokens leads to inefficient unlearning in large vocabularies by predominantly boosting other tokens; (2) **Unbounded loss**, where minimizing $\log(p_\theta(x_t|x_{<t}))$ through maximizing cross-entropy loss introduces a risk of divergence due to the unbounded nature of entropy; and (3) **Degradation of generative performance**, where GA applies uniform gradient updates (*i.e.* increasing the scores of other tokens) to all sequences in the forget set $\mathcal{D}_f$, despite each sequence requiring a unique number of updates for unlearning. This redundancy can cause degrade the model's generative capabilities, resulting in catastrophic forgetting.

Inverted Hinge Loss. To cope with aforementioned limitations of GA, we aim to design a new loss function that achieves effective unlearning by decreasing the prediction score of the true token, while focusing gradient updates on only a minimal number of viable replacements for the ground-truth token. Inspired by the Hinge Loss (Cortes & Vapnik, 1995), we devise Inverted Hinge Loss (IHL) as:

$$\mathcal{L}_{\text{IHL}}(\boldsymbol{x}) = 1 + p_\theta(x_t|x_{<t}) - \max_{v \neq x_t}(p_\theta(v|x_{<t}))$$

As the probability $p_\theta(x_t|x_{<t})$ is given by the softmax function, the derivative of $\mathcal{L}_{\text{IHL}}(\boldsymbol{x})$ with respect to $y_t^{(v)}$ is:

$$\frac{\partial \mathcal{L}_{\text{IHL}}(\boldsymbol{x})}{\partial y_t^{(v)}} = \begin{cases} p_\theta(x_t|x_{<t})(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t}) + 1) & \text{if } v = x_t \\ p_\theta(v^\star|x_{<t})(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t}) - 1) & \text{if } v = v^\star \\ p_\theta(v|x_{<t})(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t})) & \text{if } v \neq x_t \text{ and } v \neq v^\star, \end{cases}$$

where $v^\star = \arg\max_{v \neq x_t} p_\theta(v|x_{<t})$. The detailed derivation can be found in Appendix A.

The above derivative clearly illustrate how the IHL addresses the shortcomings of GA in knowledge unlearning for LLMs. First, IHL mitigates **gradient spread** by ensuring that the gradients primarily focus on the true token x_t and its competitive token $v^\star$, without excessively boosting irrelevant tokens in large vocabularies. For example, in the case where unlearning has not yet been achieved (*i.e.*, when $p_\theta(x_t|x_{<t}) > p_\theta(v^\star|x_{<t})$), the absolute value of the gradient for the true token x_t is equal to or greater than that of $v^\star$ (with opposite sign) and exceeds that of other tokens (*i.e.*, $v \neq x_t$ and $v \neq v^\star$). This ensures efficient and targeted unlearning while avoiding the unnecessary spread of gradients across irrelevant tokens. Second, IHL resolves the issue of **unbounded loss** by defining a bounded loss function, ensuring that the prediction scores for tokens other than x_t and $v^\star$ decrease once unlearning is complete (*i.e.*, when $p_\theta(x_t|x_{<t})$ becomes less than $p_\theta(v^\star|x_{<t})$). This bounded loss prevents instability and divergence during unlearning. Lastly, IHL prevents the **degradation of generative performance** by adapting gradient updates based on the prediction scores of x_t and $v^\star$, alleviating redundant updates for other tokens (*i.e.*, $v \neq x_t$ and $v \neq v^\star$). Compared to GA, this approach reduces redundant updates for other tokens in each sequence of the forget set $\mathcal{D}_f$, thereby preserving the model's generative capabilities while achieving effective and stable unlearning.

3.4 FILA: A NOVEL LORA INITIALIZATION FOR LLM UNLEARNING

Motivation. While IHL effectively stabilizes the unlearning process by reducing the likelihood of unwanted tokens in a controlled manner, we empirically find that combining IHL with LoRA naïvely leads to requiring large number of unlearning iterations to fully forget samples in $\mathcal{D}_f$. Naturally, an increasing number of updates with $\mathcal{D}_f$ comes with the risk of overfitting, which for unlearning, may result in significant loss of knowledge on $\mathcal{D}_r$. We hypothesize that this is due to the random initialization of LoRA weights together with its low-rank structure imposing too strong a regularization to handle precise gradients needed for proper minimization of IHL (Biderman et al., 2024). Drawing inspiration from PiSSA (Meng et al., 2024), we therefore aim to accelerate optimization of IHL under LoRA by extracting weights relatively more important to $\mathcal{D}_f$ than to $\mathcal{D}_r$ from each pretrained weight, then using them to initialize LoRA weights prior to unlearning. We conjecture that this approach reinforces the model's plasticity to forget $\mathcal{D}_f$ as well as its stability on keeping knowledge on $\mathcal{D}_r$. The remainder of this section presents how parameter importances are measured via Fisher information, and how low-rank adapter weights are initialized based on the measured importances.

Parameter Importances via Fisher Information. The Fisher information matrix $\boldsymbol{F}_\theta(\mathcal{D})$ is defined as the variance of the partial derivative of the log-likelihood of data $\mathcal{D}$ with respect to the model parmeter θ (left of Eq. 2). Intuitively, the matrix can be considered a measurement on how much the model output changes following a small change on its parameter weight. However, as marginalizing across the space of $\mathcal{D}$ is intractable, many works in continual learning (Kirkpatrick et al., 2017) and model compression (Hsu et al., 2022) literature have thus used the empirical Fisher information $\hat{\boldsymbol{F}}_\theta$ instead. In the context of LLMs, this can be computed as:

$$\boldsymbol{F}_\theta(\mathcal{D}) = \mathbb{E}_{\mathcal{D}}\left[\left(\frac{\partial}{\partial\theta}\log p_\theta(\mathcal{D}|\theta)\right)^2\right] \approx \frac{1}{|\mathcal{D}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\left(\frac{\partial}{\partial\theta}\mathcal{L}_{\text{LM}}(\boldsymbol{x};\theta)\right)^2 =: \hat{\boldsymbol{F}}_\theta(\mathcal{D}), \tag{2}$$

where $\mathcal{L}_{\text{LM}}$ denotes the next-token prediction loss used to pretrain LMs, $\mathcal{L}_{\text{LM}}(\boldsymbol{x};\theta) = \sum_{t=1}^{T}\log(p_\theta(x_t|x_{<t}))$. Within our LLM unlearning setup, a high empirical Fisher information measured with $\mathcal{D}_f$ indicates that $\mathcal{L}_{\text{LM}}$ on $\mathcal{D}_f$ leads to large absolute gradients on the parameter under concern, and we consider such parameters to be *important* in generating sequences in $\mathcal{D}_f$.

Let $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{f} \coloneqq \hat{\boldsymbol{F}}_{\boldsymbol{W}}(\mathcal{D}_f)$ denote the empirical Fisher information matrix of the target parameter $\boldsymbol{W}$ measured using the forget set $\mathcal{D}_f$ (*resp.* $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{r}$ using the retain set $\mathcal{D}_r$). Then, we use the relative Fisher information $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{\text{rel}} \coloneqq \hat{\boldsymbol{F}}_{\boldsymbol{W}}^{f}/\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{r} \in \mathbb{R}^{d\times r}$ as an importance metric to identify parameters that are important exclusively for $\mathcal{D}_f$ and not for $\mathcal{D}_r$. While generating $\mathcal{D}_f$ involves extracting memorized information on $\mathcal{D}_f$ as well as composing linguistically fluent outputs, we only wish to adjust parameters responsible for the former and thus use $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{\text{rel}}$ rather than $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{f}$.

Fisher-weighted Initialization of Low-rank Adapters. Given the relative importance $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{\text{rel}}$ for each target weight $\boldsymbol{W}$, we propose to initialize the corresponding LoRA adapter weights with the solution to the following Weighted Low-Rank Approximation (WLRA) problem:

$$\min_{\boldsymbol{A}\in\mathbb{R}^{r\times k},\boldsymbol{B}\in\mathbb{R}^{d\times r}} \sum_{i,j}\left([\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{\text{rel}}]_{i,j}(\boldsymbol{W}-\boldsymbol{B}\boldsymbol{A})_{i,j}\right)^2.$$

Note that when all weights $[\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{\text{rel}}]_{i,j}$ equal one, WLRA reduces to standard low-rank matrix approximation, for which the solution can easily be computed via rank-r SVD. For general weights, however, this minimization problem does not have a closed-form solution and requires iterative optimization (Srebro & Jaakkola, 2003). While we may resort to iterative methods to initialize LoRA weights in a fine-grained manner, this would undermine the efficiency gains from deploying low-rank adapters. Therefore, we assume that parameters in each row of $\boldsymbol{W}$ share the same importance equal to the square-root of the row-wise sum of $\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{\text{rel}}$, and simplify the problem to

$$\min_{\boldsymbol{A}\in\mathbb{R}^{r\times k},\boldsymbol{B}\in\mathbb{R}^{d\times r}} \left\|\texttt{diag}\left((\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{\text{rel}}\mathbb{1})^{\frac{1}{2}}\right)(\boldsymbol{W}-\boldsymbol{B}\boldsymbol{A})\right\|_2$$

with $\mathbb{1}\in\mathbb{R}^k$ and $\texttt{diag}(\cdot)$ indicating the all-one vector and the vector diagonalization function, respectively. Unlike general WLRA, this row-wise WLRA problem has a closed-form solution, which can be obtained by applying rank-r SVD to decompose $\text{diag}(\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{\text{rel}}\mathbb{1})\boldsymbol{W} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$ and computing $\boldsymbol{B}^* = (\hat{\boldsymbol{F}}_{\boldsymbol{W}}^{\text{rel}}\mathbb{1})^{-1}\boldsymbol{U}\boldsymbol{S}^{\frac{1}{2}}$ and $\boldsymbol{A}^* = \boldsymbol{S}^{\frac{1}{2}}\boldsymbol{V}^T$.

Given this solution, we use $\boldsymbol{B}^*$ and $\boldsymbol{A}^*$ as initial LoRA weights. To ensure that the model behavior remains the same after LoRA initialization, the base layers are also updated with $\boldsymbol{W}^* = \boldsymbol{W} - \boldsymbol{B}^*\boldsymbol{A}^*$. Intuitively, our Fisher-weighted Initialization of Low-rank Adapters (FILA) extracts parameters that are important for generating $\mathcal{D}_f$, but not for generating $\mathcal{D}_r$, such that LoRA tuning can be focused on erasing knowledge relevant to $\mathcal{D}_f$ while keeping information regarding $\mathcal{D}_r$.

3.5 FINAL LOSS FUNCTION FOR LLM UNLEARNING

In summary, we perform unlearning on the model $\boldsymbol{\Theta} = \boldsymbol{\theta} \cup \boldsymbol{\theta}_{\text{FILA}}$, consisted of original pretrained weights $\boldsymbol{\theta}$ and the FILA-initialized low-rank adapter weights for each linear layer $\boldsymbol{\theta}_{\text{FILA}} = \{\boldsymbol{A}_\ell^*, \boldsymbol{B}_\ell^*\}_{\ell=1}^{L}$, where L represents the number of layers tuned via LoRA. Additionally, we incorporate GD, which utilizes the auxiliary retain set $\mathcal{D}_r$. The final loss function using both the proposed IHL and FILA is defined as follows:

$$\underset{\theta_{\text{FILA}}}{\text{minimize}} \sum_{\boldsymbol{x}_r\in\mathcal{D}_f,\boldsymbol{x}_f\in\mathcal{D}_r} \mathcal{L}_{\text{IHL}}(\boldsymbol{x}_f) + \mathcal{L}_{\text{LM}}(\boldsymbol{x}_r) \tag{3}$$

In practice, training (unlearning) for the LLM model is conducted by minimizing Eq.3 through stochastic gradient descent.

Table 1: Evaluation results on reasoning and generative capabilities before and after unlearning samples from the TDEC dataset. For all runs under LoRA, we use rank 16 and GD is used as the baseline. The "+IHL" results refer to experiments where GA within GD is replaced with our proposed IHL, and "+FILA" indicates results where FILA is used to initialize LoRA. Changes in performance after unlearning are denoted in parentheses, with best results under LoRA-based unlearning in **bold**.

Model	Method	Params. (%)	Epochs	EL_{10} (%)↓	MA (%)↓	Reasoning (Acc)↑	Dialogue (F1)↑	Pile (PPL)↓
GPT-Neo 125M	Before	-	-	30.9	77.4	43.4	9.4	17.8
	GA	100.0	17.2	1.0	27.4	39.9 (-3.5)	2.6 (-6.8)	577.8 (+560.0)
	GD		4.6	0.7	24.9	42.4 (-1.0)	5.9 (-3.5)	54.2 (+36.4)
	IHL		17.2	0.7	29.2	42.3 (-1.1)	10.3 (+0.9)	18.1 (+0.3)
	LoRA	1.6	8.6	0.3	20.6	40.8 (-2.6)	2.5 (-6.9)	129.4 (+111.6)
	+ IHL		11.4	0.4	22.7	41.9 (-1.5)	6.0 (-3.4)	32.9 (+15.1)
	+ FILA		**6.0**	0.3	23.9	**42.2 (-1.2)**	**10.1 (+0.7)**	**24.0 (+6.2)**
GPT-Neo 1.3B	Before	-	-	67.6	92.2	49.8	11.5	11.5
	GA	100.0	13.8	1.9	30.4	49.7 (-0.1)	8.5 (-3.0)	15.8 (+4.3)
	GD		12.8	2.2	30.9	48.4 (-1.4)	12.7 (+1.2)	10.8 (-0.7)
	IHL		7.6	0.7	30.4	48.4 (-1.4)	12.5 (+1.0)	11.0 (-0.5)
	LoRA	0.8	19.3	1.7	31.4	45.0 (-4.8)	9.7 (-1.8)	31.8 (+20.3)
	+ IHL		20.0	1.7	44.6	47.1 (-2.7)	10.2 (-1.3)	14.9 (+3.4)
	+ FILA		**13.0**	0.5	29.6	**48.3 (-1.5)**	**12.1 (+0.6)**	**14.7 (+3.2)**
GPT-Neo 2.7B	Before	-	-	70.4	93.4	52.3	11.5	10.4
	GA	100.0	10.8	1.6	31.0	51.9 (-0.4)	11.1 (-0.4)	17.9 (+7.5)
	GD		8.0	0.7	28.3	44.0 (-8.3)	12.7 (+1.2)	17.9 (+7.5)
	IHL		6.6	0.5	29.3	51.8 (-0.5)	12.9 (+1.4)	10.7 (+0.3)
	LoRA	0.7	14.0	0.1	20.4	45.9 (-6.4)	6.7 (-4.8)	61.1 (+50.7)
	+ IHL		17.8	0.0	26.7	**49.6 (-2.7)**	8.5 (-2.6)	22.2 (+11.8)
	+ FILA		**10.3**	0.1	28.5	**49.6 (-2.7)**	**10.7 (-0.8)**	**16.0 (+5.6)**

4 EXPERIMENTS

In this section, we first perform experiments unlearning samples from the Training Data Extraction Challenge (TDEC; §4.1), followed by ablation and analytical results (§4.2). We also conduct experiments on the Task of Fictitious Unlearning (TOFU; §4.3), a benchmark that well-mimics a real-world scenario for LLM unlearning evaluation. For brevity, we present results from additional experiments such as continual unlearning in Appendix D.

4.1 TRAINING DATA EXTRACTION CHALLENGE

Experimental Setup. The Training Data Extraction Challenge (TDEC) dataset [1] consists of 20k examples from the Pile dataset (Gao et al., 2020) found to be easily extractable from a pretrained LLM. For each experiment, we randomly sample 32 sequences with 200 tokens to consist the forget set $\mathcal{D}_f$. For the retain set $\mathcal{D}_r$, we use the subset of WikiText (Merity et al., 2017) as it contains factual world knowledge that we wish to maintain after unlearning. We consider GPT-Neo 125M, 1.3B, and 2.7B pretrained on the Pile dataset as our base models, and unlearn $\mathcal{D}_f$ using five different forget sets. For this experiment, we use a fixed learning rate of 2e-4 and use LoRA adapters with rank $r = \{4, 8, 16, 32\}$. For reasons we illustrate later in §4.2, we choose to apply LoRA on query and value layers in the attention module and two linear layers within feed-forward layers.

Following previous work (Jang et al., 2023), we measure the unlearning efficacy with two metrics. The **n-gram Extraction Likelihood (EL_n)** measures the n-gram overlap between the ground truth sequence in $\mathcal{D}_f$ and the output generated by the model. The **Memorization Accuracy (MA)** measures the token-wise accuracy of the LLM on $\mathcal{D}_f$. More details on these metrics are shared in Appendix B. After each unlearning epoch, we measure EL_{10} and MA of the model, and we consider the model has successfully unlearned $\mathcal{D}_f$ if both values measured on $\mathcal{D}_f$ become smaller than those measured from a held-out validation set that the model has never seen before within 20 unlearning epochs. Once unlearning is finished, we evaluate the unlearned model on various downstream benchmarks to measure how well the LLM maintains its previously acquired reasoning and generative capabilities. To assess its reasoning capabilities, we average accuracies across 9 different classification datasets. To measure generative performance, we also average the F1 scores over four dialogue generation datasets. Lastly, we measure the perplexity on the validation subset of the Pile (Gao et al., 2020). A comprehensive list of evaluation datasets can be found in Appendix C.

---

[1] The dataset was originally published as part of a competition held at SaTML 2023: `https://github.com/google-research/lm-extraction-benchmark`

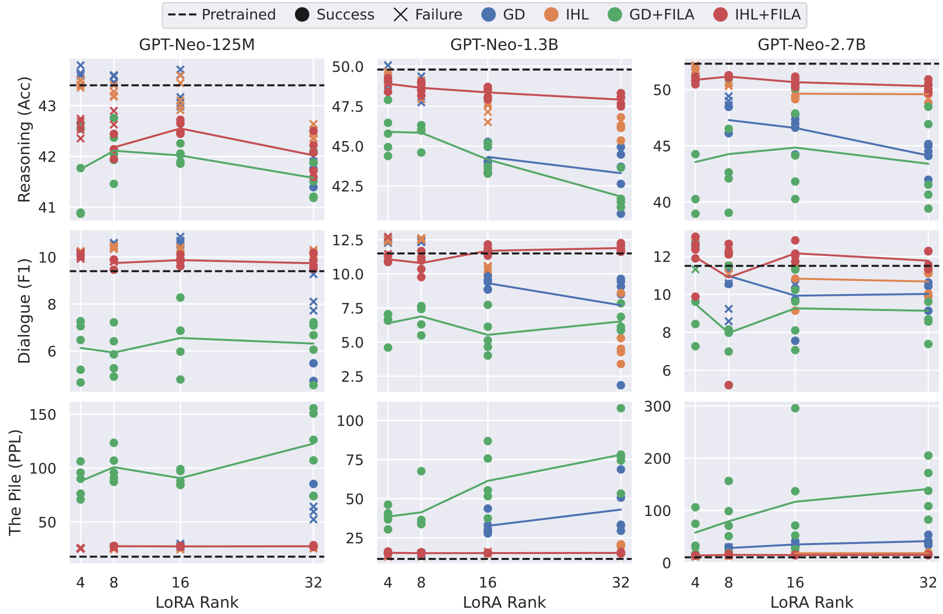


Figure 3: Results from unlearning examples in the TDEC dataset on the GPT-Neo LLM family. Each row represents the performance averaged across datasets within each set of LLM capability tests: Reasoning (higher is better), Dialogue (higher is better), and Perplexity (lower is better). The circles and crosses represent successful and unsuccessful attempts, respectively, of unlearning a particular forget set $\mathcal{D}_f$. Solid lines indicate the performance of different methods averaged *only across successful unlearning trials*. The dashed lines indicate the base model performance prior to unlearning. GD leads to significant loss in performance and also fails to unlearn in some cases even with large LoRA ranks. Replacing the NCE loss in GD with IHL boosts retention of reasoning and generation capabilities, but still fails to unlearn in multiple cases. Running GD with FILA notably increases the rate of unlearning success, but at significant cost in overall performance. Using both IHL and FILA best minimizes post-unlearning performance degradation in all three aspects.

We consider two LLM unlearning baselines, Gradient Ascent (GA) (Jang et al., 2023) and Gradient Difference (GD) (Liu et al., 2022a; Maini et al., 2024), both of which only require the original language model and datasets $\mathcal{D}_f$ and $\mathcal{D}_r$ representing knowledge we wish to unlearn and retain, respectively. We exclude methods that require another auxiliary model (Wang et al., 2023; Liu et al., 2024) or the entire training data (Wang et al., 2023; Chen & Yang, 2023) from our baselines.

Results. Table 1 shows evaluation results using a fixed LoRA rank of 16 and Figure 3 shows analogous results using a different LoRA ranks. Our key findings are as follows. First, using GD not only meets the forgetting criteria in fewer epochs across all model sizes but it also preserves previously acquired knowledge (*i.e.*, performance on Reasoning, Dialogue, and Pile) better than GA. Mainly for 125M and 1.3B models, GA causes a significant decline in generative performance, whereas GD partially mitigates this decline and improves both Dialogue F1 scores and the Pile perplexity. Second, we find that simply applying GD with LoRA fails to unlearn effectively across all model sizes. While enjoying great parameter-efficiency by tuning only about 0.7% to 1.6% of the total parameters when using LoRA, its application to unlearning with GD results in large loss in overall language capability, especially on generative tasks. Third, replacing GA in GD with IHL leads to performance gains in both full-parameter and LoRA-based unlearning, but requires larger number of epochs for successful unlearning than GD especially when confined to low-rank weight changes. However, this is resolved when IHL is used together with FILA, which significantly reduces the number of required epochs. Using our proposed IHL in conjunction with FILA achieves the forgetting criteria in fewest epochs while best preserving LLM performance.

4.2 ANALYSIS

What modules do we need to adapt? Figure 4 presents experiments where low-rank adapters are attached to various target parameter groups, including those for Query (Q), Value (V), Key (K), Output (O) in the attention module, and the Feed-Forward Network (FFN). While the original LoRA paper (Hu et al., 2022) indicates that applying LoRA to Q and V yields superior performance on downstream tasks, our experiments indicate that using LoRA on Q and V only is insufficient to meet the unlearning criteria within our timeframe of 20 epochs. Notably, when LoRA is applied to FFNs,

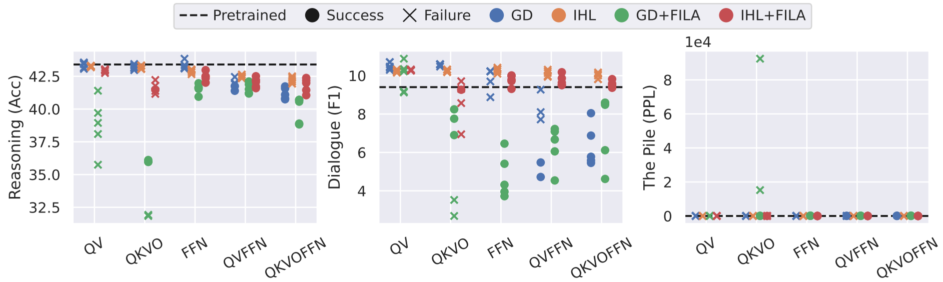


Figure 4: Results from unlearning examples from TDEC dataset using LoRA with rank 32 to adapt sets of layers on GPT-Neo-125M. The marker shapes and colors are used similarly as in Figure 3. Based on the rate of unlearning success, tuning FFN layers (*e.g.*, FFN, QVFFN) is more receptive to targeted knowledge removal compared to tuning attention layers (*e.g.*, QV, QKVO).

we observe significant increase in rate of successful unlearning. Furthermore, integrating FILA with IHL achieves the best post-unlearning performance across all LoRA target module combinations.

Cost-efficiency of the proposed method. Our compute-cost vs. performance comparisons in Figure 2 show that, while vanilla LoRA allows significant reduction in unlearning costs (*i.e.*, FLOPs) by freezing the majority of parameters, it incurs substantial performance losses compared to full-parameter unlearning due to excessive stability originating from its low-rankness. In contrast, combining the proposed IHL with FILA not only achieves the best performance but also leverages the cost advantages of LoRA.

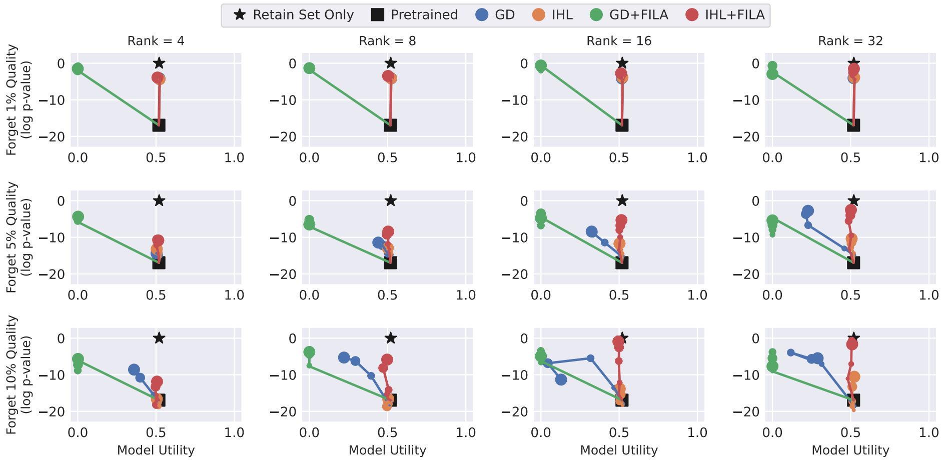
4.3 TASK OF FICTITIOUS UNLEARNING

Experimental Setup. The Task of Fictitious Unlearning (TOFU) benchmark (Maini et al., 2024) is a synthetic dataset containing 20 question-answer pairs for each of 200 fictitious author profiles generated by GPT-4. The TOFU evaluation pipeline first finetunes a pretrained LLM on all QA pairs. Given this finetuned LLM that serves as our base model, our task is to unlearn all information regarding 1%, 5%, or 10% of the authors from the model. Note that we can obtain reference models finetuned only on the retain set (QA-pairs on 99%, 95%, or 90% of authors), with which we evaluate the **Forget Quality** of unlearned models by measuring the p-value from a Kolmogorov-Smirnov test. A high p-value indicates high distributional similarity between the unlearned model and the reference model, thus implying strong forgetting. To evaluate how well the model retains other information outside the forget set, we measure the **Model Utility** as the aggregation of the probability of correct answers, ROUGE-L scores, and Truth Ratio of correct answers vs. incorrect answers on questions from three datasets pertaining to the retain set of fictitious authors, real authors, and world facts.

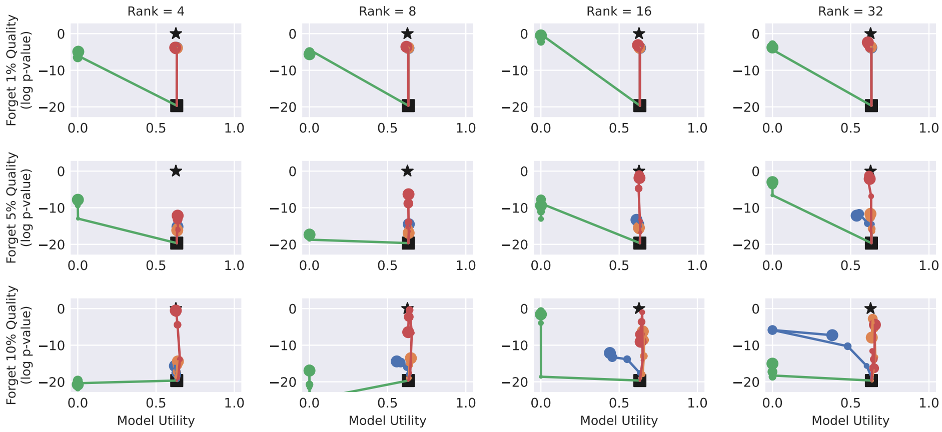
Following the original paper of TOFU, we prepare two base models by finetuning Phi-1.5B and Llama2-7B on TOFU for 5 epochs with learning rates 2e-5 and 1e-5, respectively. We then unlearn with various methods using LoRA adapters of rank 4, 8, 16, or 32. While we mainly compare our methods against GD, we also compare IHL and FILA against existing unlearning methods such as KL (Maini et al., 2024), DPO (Rafailov et al., 2024), and NPO (Zhang et al., 2024), results from which can be found in Appendix D. For unlearning, we use a learning rate of 2e-4 if our base model is from Phi-1.5B and 1e-4 for Llama2-7B. All training procedures run 5 epochs with an effective batch size of 32 using the AdamW optimizer (Loshchilov & Hutter, 2019).

Results. Figure 5 shows the model utility vs. forget quality curves from unlearning three differently-sized TOFU forget sets from Phi-1.5B and Llama2-7B models. Comparing results among different forget set sizes, we first observe that forgetting 1% of author profiles is fairly straightforward, as all curves quickly approach the reference model with a single epoch, with increasing the LoRA rank leading to incremental improvements in performance. On the other hand, when unlearning a larger set of profiles (*i.e.*, 5% or 10%), we see that both GA and GD quickly degrades model utility.

With regards to our proposed method, we find that replacing the NCE loss in GD with our IHL better retains model utility across all LoRA ranks and forget set sizes, as curves are more aligned straight-up towards the reference point with negligible shift in model utility. This stability comes at the cost of unlearning efficiency, however, as randomly initialized LoRA weights are unable to effectively represent weight changes required to decrease IHL. Nonetheless, initializing LoRA adapters with FILA largely alleviates this issue and significantly enhances unlearning efficiency of IHL by focusing gradient updates on parameters important to generating $\mathcal{D}_f$.



(a) Phi-1.5B



(b) Llama2-7B

Figure 5: TOFU results using Phi-1.5B and Llama2-7B models. Each row corresponds to unlearning a different forget set (1%, 5%, or 10%), and each column uses a distinct LoRA rank between 4 and 32. The relative size of markers represent the number of epochs. Ideally, the unlearning curves should start from the pretrained model (■) and approach towards the reference model tuned on the retain set only (★) as unlearning progresses. Both GD and GD+FILA suffers from significant loss of model utility due to using GA for unlearning. Replacing GA with IHL largely retains model utility, then initializing LoRA adapters with FILA significantly boosts the unlearning efficiency of IHL.

Interestingly, we find the prior weight assignment via FILA can lead to excessive unlearning in some cases (*e.g.*, unlearning 10% forget set with ranks 8 or 16 on Llama2-7B), with model updates reducing the forget quality after reaching the upper bound at zero. This behavior resembles the *Streisand effect* as unlearning gradients beyond a certain point in optimization unintentionally renders $\mathcal{D}_f$ more noticeable within the model (Golatkar et al., 2020). As reference models are not available for measuring forget quality in real-world scenarios, finding the optimal point at which to stop unlearning to prevent this effect as well as designing a robust evaluation metric that does not depend upon oracle models would be interesting directions, which we leave as future work.

5 CONCLUDING REMARKS

In this paper, we address limitations of Gradient Ascent (GA), a widely used method for LLM unlearning, and introduce a novel Inverted Hinge Loss (IHL) to replace the negative cross-entropy loss in GA and resolve issues with dispersed gradients and unboundedness. We also propose Fisher-weighted initialization for low-rank adaptation (FILA) that pre-assigns weights relatively important to generating unwanted information as means to facilitate efficient LLM unlearning with LoRA. Experiments on the Training Data Extraction Challenge dataset with GPT-Neo models along with the TOFU benchmark using Phi-1.5B and Llama2-7B models show that our proposed methods enable faster and more stable LoRA-based LLM unlearning, significantly outperforming existing baselines in computational efficiency as well as post-unlearning performance.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7319–7328, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.568. URL `https://aclanthology.org/2021.acl-long.568`.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2357–2367, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1245. URL `https://aclanthology.org/N19-1245`.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=aloEru2qCG`. Featured Certification.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–480. IEEE, 2015.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=TatRHT_1cK`.

Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11186–11194, 2024.

Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12041–12052, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.738. URL `https://aclanthology.org/2023.emnlp-main.738`.

Kongyang Chen, Zixin Wang, Bing Mi, Waixi Liu, Shaowei Wang, Xiaojun Ren, and Jiaxing Shen. Machine unlearning in large language models. *arXiv preprint arXiv:2404.16841*, 2024.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i6.25879. URL `https://doi.org/10.1609/aaai.v37i6.25879`.

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

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=r1l73iRqKm`.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Chongyang Gao, Lixu Wang, Chenkai Weng, Xiao Wang, and Qi Zhu. Practical unlearning for large language models. *arXiv preprint arXiv:2407.10223*, 2024.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9304–9312, 2020.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In Eneko Agirre, Johan Bos, Mona Diab, Suresh Manandhar, Yuval Marton, and Deniz Yuret (eds.), **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pp. 394–398, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL `https://aclanthology.org/S12-1052`.

Michael M Grynbaum and Ryan Mac. The times sues openai and microsoft over ai use of copyrighted work. *The New York Times*, 27, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. Language model compression with weighted low-rank factorization. 2022. URL `https://openreview.net/forum?id=uPv9Y3gmAI5`.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=6t0Kwf8-jrj`.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.805. URL `https://aclanthology.org/2023.acl-long.805`.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1259. URL `https://aclanthology.org/D19-1259`.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. Internet-augmented dialogue generation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8460–8478, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.579. URL `https://aclanthology.org/2022.acl-long.579`.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. VeRA: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=NjNfLdxr3A`.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=ryup8-WCW`.

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

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022b.

Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, et al. Parameter-efficient orthogonal finetuning via butterfly factorization. *arXiv preprint arXiv:2311.06243*, 2023.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Towards safer large language models through machine unlearning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 1817–1829, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.107. URL `https://aclanthology.org/2024.findings-acl.107`.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL `https://arxiv.org/abs/1711.05101`.

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

Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N Ravi. Deep unlearning via randomized conditionally independent hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10422–10431, 2022.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *arXiv preprint arXiv:2404.02948*, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=Byj72udxe`.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL `https://aclanthology.org/P16-1144`.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. *Advances in Neural Information Processing Systems*, 36:79320–79362, 2023.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

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

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. Towards empathetic open-domain conversation models: A new benchmark and dataset. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5370–5381, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1534. URL `https://aclanthology.org/P19-1534`.

Jeffrey Rosen. The right to be forgotten. *Stan. L. Rev. Online*, 64:88, 2011.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Nianwen Si, Hao Zhang, Heyu Chang, Wenlin Zhang, Dan Qu, and Weiqiang Zhang. Knowledge unlearning for llms: Tasks, methods, and challenges. *arXiv preprint arXiv:2311.15766*, 2023.

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. Can you put it all together: Evaluating conversational agents' ability to blend skills. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2021–2030, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.183. URL `https://aclanthology.org/2020.acl-main.183`.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*, 2022.

Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 720–727, 2003.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

Eduard Fosch Villaronga, Peter Kieseberg, and Tiffany Li. Humans forget, machines remember: Artificial intelligence and the right to be forgotten. *Computer Law & Security Review*, 34(2):304–313, 2018.

Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. KGA: A general machine unlearning framework based on knowledge gap alignment. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13264–13276, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.740. URL `https://aclanthology.org/2023.acl-long.740`.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. DEPN: Detecting and editing privacy neurons in pretrained language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2875–2886, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.174. URL `https://aclanthology.org/2023.emnlp-main.174`.

Shih yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=3d5CIRG1n2`.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8403–8419, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.457. URL `https://aclanthology.org/2024.acl-long.457`.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8403–8419, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.457. URL `https://aclanthology.org/2024.acl-long.457`.

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

Shih-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard BW Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. In *The Twelfth International Conference on Learning Representations*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL `https://aclanthology.org/P19-1472`.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

A DERIVATIVE ANALYSIS FOR THE INVERTED HINGE LOSS FUNCTION

The function $p_\theta(x_t|x_{<t})$ represents a probability distribution that indicates the likelihood of x_t taking a specific token x_t given the previous tokes $x_{<t}$. This probability is expressed using the softmax function: $p_\theta(x_t|x_{<t}) = \exp(y_t^{(x_t)})/\sum_{v=1}^{V}\exp(y_t^{(v)})$, where $y_t^{(v)}$ denotes the score for the v-th token in the vocabulary. To differentiate this function with respect to $y_t^{(x_t)}$, we rewrite $p_\theta(x_t|x_{<t}) = \exp(y_t^{(x_t)})/Z$ where $Z = \sum_{v=1}^{V}\exp(y_t^{(v)})$ is the normalization constant.

We differentiate this function with respect to $y_t^{(k)}$ considering two cases: 1) $k = x_t$ and 2) $k \neq x_t$. For the first case, we can get the following by using the chain rule:

$$\frac{\partial p_\theta(x_t|x_{<t})}{\partial y_t^{(x_t)}} = \frac{\partial}{\partial y_t^{(x_t)}}\left(\frac{\exp(y_t^{(x_t)})}{Z}\right) = \frac{1}{Z}\frac{\partial \exp(y_t^{(x_t)})}{\partial y_t^{(x_t)}} - \frac{\exp(y_t^{(x_t)})}{Z^2}\frac{\partial Z}{\partial y_t^{(x_t)}}$$

Here, $\frac{\partial \exp(y_t^{(x_t)})}{\partial y_t^{(x_t)}} = \exp(y_t^{(x_t)})$ and $\frac{\partial Z}{\partial y_t^{(x_t)}} = \exp(y_t^{(x_t)})$. Therefore, it becomes:

$$\frac{\partial p_\theta(x_t|x_{<t})}{\partial y_t^{(x_t)}} = \frac{\exp(y_t^{(x_t)})}{Z} - \frac{\exp(y_t^{(x_t)})^2}{Z^2} = p_\theta(x_t|x_{<t}) - p_\theta(x_t|x_{<t})^2 = p_\theta(x_t|x_{<t})(1 - p_\theta(x_t|x_{<t}))$$

For the second case, using the chain rule again, we get:

$$\frac{\partial p_\theta(x_t|x_{<t})}{\partial y_t^{(k)}} = \frac{\partial}{\partial y_t^{(k)}}\left(\frac{\exp(y_t^{(x_t)})}{Z}\right) = -\frac{\exp(y_t^{(x_t)})}{Z^2}\frac{\partial Z}{\partial y_t^{(k)}}$$

where $\frac{\partial Z}{\partial y_t^{(k)}} = \exp(y_t^{(k)})$. Therefore,

$$\frac{\partial p_\theta(x_t|x_{<t})}{\partial y_t^{(k)}} = -\frac{\exp(y_t^{(x_t)})\exp(y_t^{(k)})}{Z^2} = -p_\theta(x_t|x_{<t}) \cdot p_\theta(k|x_{<t})$$

Thus, we can summarize them as below:

$$\frac{\partial p_\theta(x_t|x_{<t})}{\partial y_t^{(v)}} = \begin{cases} p_\theta(x_t|x_{<t})(1 - p_\theta(x_t|x_{<t})) & \text{if } v = x_t \\ -p_\theta(x_t|x_{<t}) \cdot p_\theta(v|x_{<t}) & \text{if } v \neq x_t \end{cases}$$

Based on the derivative of $p_\theta(x_t|x_{<t})$ above , we can calculate the derivative of $\mathcal{L}_{\text{IHL}}$. Firstly, for convenience, we define $p_t = p_\theta(x_t|x_{<t})$ and $\hat{p}_t = \max_{v\neq x_t}(p_\theta(v|x_{<t}))$. The loss function can be rewritten as:

$$\mathcal{L}_{\text{IHL}}(\boldsymbol{x}) = 1 + p_t - \hat{p}_t$$

To calculate the derivative of $\mathcal{L}_{\text{IHL}}$, we need to consider three cases: 1) when $v = x_t$, 2) when $v = v^\star$ where $v^\star = \arg\max_{v\neq x_t} p_\theta(v|x_{<t})$, 3) when $v \neq x_t$ and $v \neq v^\star$. Using the derivative of $p_\theta(x_t|x_{<t})$ mentioned earlier, the derivative of $\mathcal{L}_{\text{IHL}}$ with respect to $y_t^{(v)}$ is as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{IHL}}}{\partial y_t^{(x_t)}} &= \frac{\partial}{\partial y_t^{(x_t)}}\left(1 + p_\theta(x_t|x_{<t}) - p_\theta(v^\star|x_{<t})\right) \\ &= p_\theta(x_t|x_{<t})(1 - p_\theta(x_t|x_{<t})) + p_\theta(x_t|x_{<t}) \cdot p_\theta(v^\star|x_{<t}) \\ &= p_\theta(x_t|x_{<t})\left(1 - p_\theta(x_t|x_{<t}) + p_\theta(v^\star|x_{<t})\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{IHL}}}{\partial y_t^{(v^\star)}} &= \frac{\partial}{\partial y_t^{(v^\star)}}\left(1 + p_\theta(x_t|x_{<t}) - p_\theta(v^\star|x_{<t})\right) \\ &= -p_\theta(x_t|x_{<t}) \cdot p_\theta(v^\star|x_{<t}) - p_\theta(v^\star|x_{<t})(1 - p_\theta(v^\star|x_{<t})) \\ &= -p_\theta(v^\star|x_{<t})\left(1 - p_\theta(v^\star|x_{<t}) + p_\theta(x_t|x_{<t})\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{IHL}}}{\partial y_t^{(v)}} &= \frac{\partial}{\partial y_t^{(v)}}\left(1 + p_\theta(x_t|x_{<t}) - p_\theta(v^\star|x_{<t})\right) \\ &= -p_\theta(x_t|x_{<t}) \cdot p_\theta(v|x_{<t}) + p_\theta(v^\star|x_{<t}) \cdot p_\theta(v|x_{<t}) \\ &= p_\theta(v|x_{<t})\left(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t})\right)\end{aligned}$$

In summary, the derivatives of the loss function $\mathcal{L}_{\text{IHL}}$ with respect to $y_t^{(v)}$ for the three cases are:

$$\frac{\partial \mathcal{L}_{\text{IHL}}(\boldsymbol{x})}{\partial y_t^{(v)}} = \begin{cases} p_\theta(x_t|x_{<t})(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t}) + 1) & \text{if } v = x_t \\ p_\theta(v^\star|x_{<t})(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t}) - 1) & \text{if } v = v^\star \\ p_\theta(v|x_{<t})(p_\theta(v^\star|x_{<t}) - p_\theta(x_t|x_{<t})) & \text{if } v \neq x_t \text{ and } v \neq v^\star, \end{cases}$$

B EVALUATION METRICS

How to measure success of unlearning? Following previous work Jang et al. (2023); Tirumala et al. (2022), we empirically measure the success of unlearning using two metrics, Extraction Likelihood (EL) and Memorization Accuracy (MA), which we briefly discuss below.

After unlearning each sequence $\boldsymbol{x} = (x_1, \ldots, x_T) \in \mathcal{D}_f$, the Extraction Likelihood (EL) is measured as the n-gram overlap between the ground truth sequence $\boldsymbol{x}$ and the output of the model after unlearning.

$$\textsc{Overlap}_n(\boldsymbol{a}, \boldsymbol{b}) = \frac{\sum_{\boldsymbol{c} \in n\text{-GRAM}(\boldsymbol{a})} \mathbb{1}\{\boldsymbol{c} \in n\text{-GRAM}(\boldsymbol{b})\}}{|n\text{-GRAM}(\boldsymbol{a})|} \tag{4}$$

$$\text{EL}_n(\boldsymbol{x}) = \frac{\sum_{t=1}^{T-n} \textsc{Overlap}_n\left(f_\theta(x_{<t}), x_{\geq t}\right)}{T - n} \tag{5}$$

The Memorization Accuracy (MA) measures the token-wise memorization of the LM p_θ.

$$\text{MA}(\boldsymbol{x}) = \frac{\sum_{t=1}^{T} \mathbb{1}\{\arg\max_x p_\theta(x|x_{<t}) = x_t\}}{T - 1} \tag{6}$$

Given these two metrics, we flag successful unlearning when the average EL and MA on $\mathcal{D}_f$ goes below the EL and MA values measured on the validation set unseen during training. In our experiments we measure EL with 10-grams, which results in the following early stopping criterion.

$$\frac{1}{\mathcal{D}_f} \sum_{\boldsymbol{x} \in \mathcal{D}_f} \text{EL}_{10}(\boldsymbol{x}) \leq \frac{1}{\mathcal{D}_{\text{val}}} \sum_{\boldsymbol{x} \in \mathcal{D}_{\text{val}}} \text{EL}_{10}(\boldsymbol{x}) \quad \text{and} \quad \frac{1}{\mathcal{D}_f} \sum_{\boldsymbol{x} \in \mathcal{D}_f} \text{MA}(\boldsymbol{x}) \leq \frac{1}{\mathcal{D}_{\text{val}}} \sum_{\boldsymbol{x} \in \mathcal{D}_{\text{val}}} \text{MA}(\boldsymbol{x})$$

C ADDITIONAL DETAILS ON EXPERIMENTAL SETTING

Experiemtal Settings All experiments were conducted on a remote server equipped with NVIDIA A100 40GB Tensor Core GPUs.

Datasets for Evaluation in the TDEC To evaluate reasoning capabilities, we utilize nine different classification datasets: LAMBADA (Paperno et al., 2016), Hellaswag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2021), COPA (Gordon et al., 2012), ARC-Easy (Clark et al., 2018), ARC-Challenge (Clark et al., 2018), PiQA (Bisk et al., 2020), MathQA (Amini et al., 2019), and PubmedQA (Jin et al., 2019). To assess generative performance, we employ Blended Skill Talk (Smith et al., 2020), Empathetic Dialogues (Rashkin et al., 2019), Wizard of Internet (Komeili et al., 2022), and Wizard of Wikipedia (Dinan et al., 2019).

Details of metrics of TOFU We evaluate the **Forget Quality** of unlearned models by measuring the *p*-value from the Kolmogorov-Smirnov test that compares the empirical distribution of our unlearned model to that of the reference model. To evaluate how well the model retains other information outside the forget set, we measure the **Model Utility** as the aggregated model performance on the retain set of remaining fictitious author profiles, and two held-out sets consisted of QA-pairs regarding real author profiles and other world facts.

D ADDITIONAL EXPERIMENTAL RESULTS

D.1 CONTINUAL UNLEARNING

Because of the importance of continual unlearning (or sequential unlearning) in real-world applications, previous studies have underscored its relevance through a sequence of unlearning tasks (Cha et al., 2024; Jang et al., 2023). Building on them, we conduct continual unlearning experiments involving four tasks. Figure 6 of the Appendix shows that IHL consistently outperforms GD across all metrics. Notably, the proposed IHL demonstrates significantly enhanced performance on the four Dialogue and Pile datasets. Finally, we confirm that the combination of IHL and FLoRA achieves more robust and cose-efficient continual unlearning, as evidenced by the experimental results for Reasoning, Dialogue, and Pile, while utilizing only about 1.6% of the total parameters.

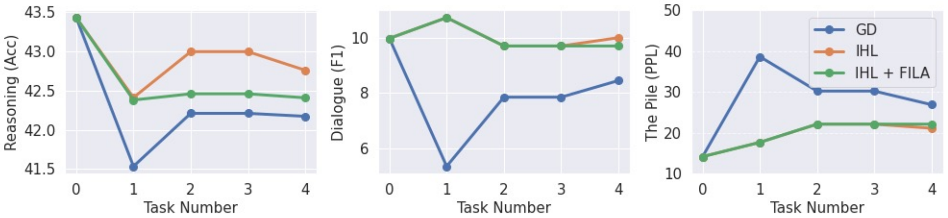
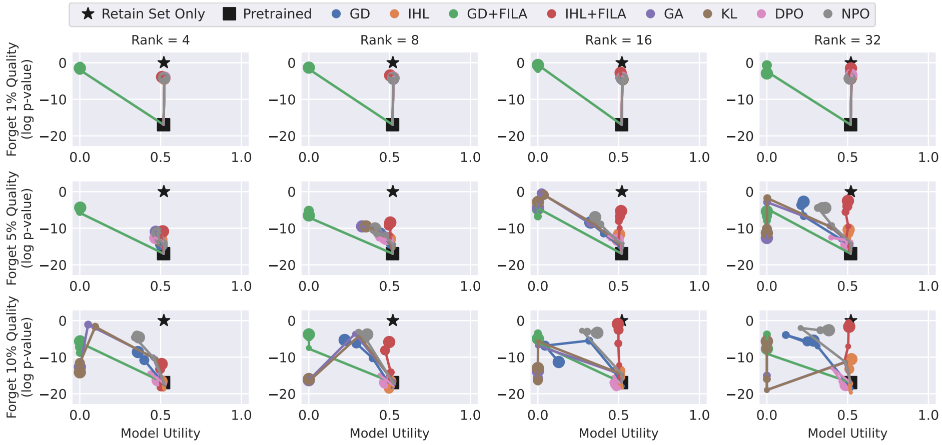


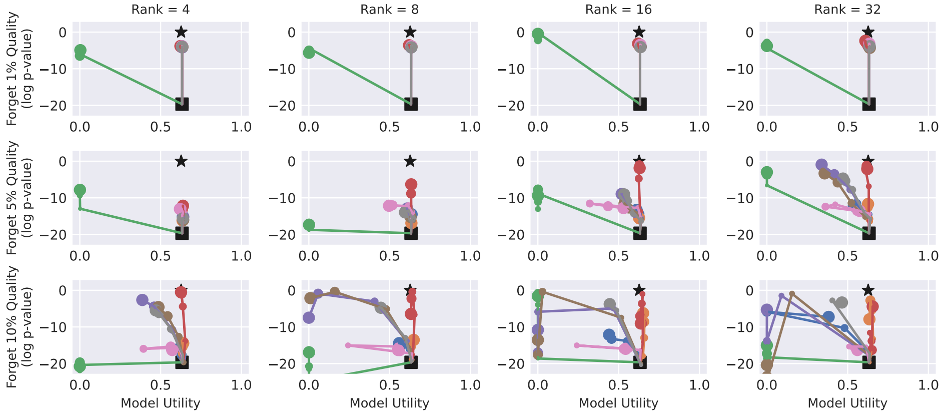
Figure 6: Experimental results of continual unlearning. Each task consists of 32 disjoint sequences sampled from the TDEC dataset, leading to a total of 128 sequences to unlearn. For these experiments, we use the pretrained GPT-Neo 125M model. The experimental setup for unlearning and the forgetting criteria are configured as in the previous TDEC experiments. Task 0 refers to the result before unlearning.

D.2 TOFU RESULTS WITH ADDITIONAL BASELINES

We compare our IHL and FILA methods against three additional existing unlearning baselines: KL (Maini et al., 2024) uses GA to forget samples in $\mathcal{D}_f$ while minimizing the Kullback-Leibler (KL) divergence between representations of retain samples in $\mathcal{D}_r$ output by the unlearned model and those from the pretrained base model. Instead of GA, Direct Preference Optimization (DPO; Rafailov et al. (2024)) performs unlearning by training the model to output variants of “I don’t know” when given a sample in $\mathcal{D}_f$. NPO (Zhang et al., 2024) is an approach similar to GA, but with adaptive weighting on the gradients such that it alleviates the divergent behavior of GA. Note that both DPO and NPO are regularized by the LM-loss on $\mathcal{D}_r$. We use the same hyperparameterization (*e.g.,* learning rate and effective batch size) as in our main results. Figure 7 shows the results. We find that all three baselines lead to significant decrease in model utility, while IHL+FILA shows negligible change.



(a) Phi-1.5B



(b) Llama2-7B

Figure 7: TOFU results using Phi-1.5B and Llama2-7B models. Each row corresponds to unlearning a different forget set (1%, 5%, or 10%), and each column uses a distinct LoRA rank between 4 and 32. The relative size of markers represent the number of epochs. Ideally, the unlearning curves should start from the pretrained model (■) and approach towards the reference model tuned on the retain set only (★) as unlearning progresses. Our method IHL+FILA outperforms existing KL- and preference optimization-based unlearning approaches in both model utility and forget quality.