A General Framework to Enhance Fine-tuning-based LLM Unlearning

Jie Ren[1], **Zhenwei Dai**[2], **Xianfeng Tang**[2], **Hui Liu**[2], **Jingying Zeng**[2], **Zhen Li**[2], **Rahul Goutam**[2], **Suhang Wang**[3], **Yue Xing**[1], **Qi He**[2], **Hui Liu**[1]
[1]Michigan State University, [2]Amazon, [3]The Pennsylvania State University
{renjie3, xingyue1, liuhui7}@msu.edu
{zwdai, xianft, liunhu, zejingyi, amzzhn, rgoutam, qih}@amazon.com
szw494@psu.edu

Abstract

Unlearning has been proposed to remove copyrighted and privacy-sensitive data from Large Language Models (LLMs). Existing approaches primarily rely on fine-tuning-based methods, which can be categorized into gradient ascent-based (GA-based) and suppression-based methods. However, they often degrade model utility (the ability to respond to normal prompts). In this work, we aim to develop a general framework that enhances the utility of fine-tuning-based unlearning methods. To achieve this goal, we first investigate the common property between GA-based and suppression-based methods. We unveil that GA-based methods unlearn by distinguishing the target data (i.e., the data to be removed) and suppressing related generations—essentially the same strategy employed by suppression-based methods. Inspired by this finding, we introduce Gated Representation UNlearning (GRUN) which has two components: a soft gate function for distinguishing target data and a suppression module using Representation Fine-tuning (ReFT) to adjust representations rather than model parameters. Experiments show that GRUN significantly improves the unlearning and utility. Meanwhile, it is general for fine-tuning-based methods, efficient and promising for sequential unlearning. Our code is available at github.com/renjie3/GRUN.

1 Introduction

LLMs have shown remarkable capabilities across various tasks (Achiam et al., 2023; Touvron et al., 2023). A key factor driving the rapid advancement is the availability of web-scale datasets. However, concerns have been raised regarding the use of such large-scale data, as it often includes copyrighted and privacy-protected data (Hacker et al., 2023; Lucchi, 2024). For instance, The New York Times sued OpenAI and Microsoft because their articles have been used in training GPT[1]. Meanwhile, the data is protected by General Data Protection Regulation (GDPR) (Voigt and Von dem Bussche, 2017), and the data owners have the "right to be forgotten" (Rosen, 2011). Therefore, it is crucial to implement protections for these datasets. To address this, unlearning has been proposed to remove specific data from LLMs without requiring full retraining (Liu et al., 2024a,b). The goal is to eliminate the influence of the **target data** or adjust the model behavior as if it had never encountered the target data.

LLM unlearning is typically a post-training method, with fine-tuning being widely adopted as an approach. Existing fine-tuning based unlearning methods can be roughly divided into two categories. One is **gradient ascent-based** (GA-based) methods, such as gradient ascent (GA) (Jang et al., 2023; Maini et al., 2024) and its variants (Yao et al., 2023; Liu et al., 2022; Zhang et al., 2024a; Fan et al., 2024; Veldanda et al., 2024; Cha et al., 2024; Liu et al., 2024c; Feng et al., 2024; Bu et al., 2024; Tian et al., 2024). They negate the training impact of the target data by reversing the gradient descent loss. The other, **suppression-based** unlearning, does not aim to erase learned information directly (Maini et al., 2024; Li et al., 2024; Wang et al., 2024c; Huu-Tien et al., 2024; Shi et al., 2024a; Liu et al., 2024c; Sinha et al., 2024). Instead, it explicitly tells the model about what constitutes target data and guides it to generate human-preferred outputs while suppressing those related to the target data [2].

However, recent evaluations on fine-tuning-based methods reveal that there is a significant trade-off between unlearning and model utility, i.e.,

[1]https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html

[2]In addition to fine-tuning, other methods operating at the inference stage have also been proposed, such as in-context learning (ICL) (Pawelczyk et al.) and assistant models (Huang et al., 2024). Nonetheless, given the widespread adoption of fine-tuning, we focus on fine-tuning methods in this work.

Findings of the Association for Computational Linguistics: ACL 2025, pages 18464–18476
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

the model's ability to respond to normal prompts unrelated to the target data (Wang et al., 2024a; Si et al., 2023; Wu et al., 2024a). This issue has been widely observed in LLM fine-tuning: as the fine-tuning dataset is small, it is likely to cause overfitting and reduce the general ability (Luo et al., 2023; Zhai et al., 2023; Howard and Ruder, 2018). Although they usually use retaining dataset to preserve the model utility (Liu et al., 2022; Shi et al., 2024b), its small size could limit the generalization.

Therefore, we aim to develop a general framework to enhance the utility of fine-tuning-based LLM unlearning. However, the two types of fine-tuning-based methods are defined in totally different ways, posing a challenge in developing such a framework. Thus, we design a preliminary study to investigate the common property between GA-based and suppression-based methods (Section 3). We find that, although GA-based methods appear to be dedicated to negate the training of target data, the final GA-unlearned LLMs still recognize target data and actually treat it as a signal of unlearning. If target data is in the input, the representations exhibit a distinct pattern compared with the input irrelevant to target data. Then unlearned models suppress related generation. This suggests that the GA-unlearned models also operate by distinguishing and suppressing target data, which closely resemble the models by suppression-based methods.

Inspired by the insights from our preliminary study that both GA-based and suppression-based methods rely on distinguishing target data for unlearning, we introduce **G**ated **R**epresentation **UN**learning (GRUN). GRUN consists of two plug-and-play components designed explicitly for distinguishing and suppression: a soft gate function to distinguish, and a suppression module utilizing Representation Fine-Tuning (ReFT) (Wu et al., 2024b). The ReFT module fine-tunes the representation instead of the model parameters, which can avoid distorting the parameters to preserve the utility. Meanwhile, its strength is controlled by the soft gate function, which further ensures the generation unrelated to the target data remains almost untouched. In essence, the soft gate function selectively activates for target data, while the ReFT module unlearns by redirecting the embeddings of target-data-related prompts toward suppression.

We conduct extensive experiments to examine the effectiveness and efficiency of GRUN. GRUN requires a lightweight additional module (less than 0.05 % of the LLM's size) and reduces training time by over 95% compared to the original method, yet achieves near-perfect unlearning while maintaining utility. Moreover, GRUN is a general solution adaptable to various fine-tuning-based unlearning methods. Our experiments validate this across various models, including Llama 3.1 and Mistral, as well as across different datasets, such as TOFU focusing on the unlearning of fine-tuning data (Maini et al., 2024), and WMDP focusing on unlearning pre-training data (Li et al., 2024).

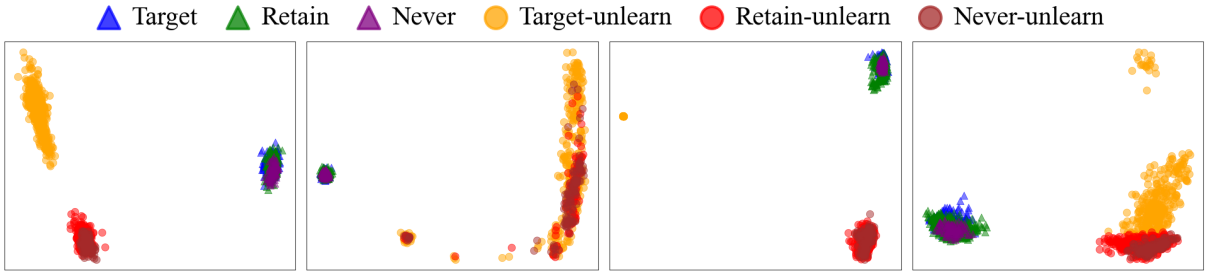
2 Related works

LLM unlearning. Machine unlearning focused on vision models in the early research (Cao and Yang, 2015; Warnecke et al., 2021; Bourtoule et al., 2021; Kurmanji et al., 2024; Ren et al., 2024; Li et al., 2021), but more recently, it has been extended to LLMs (Eldan and Russinovich, 2023; Yao et al., 2023; Shi et al., 2024b; Liu et al., 2024b). Fine-tuning-based methods represent a key category of unlearning but raise concerns regarding their impact on model utility (Thaker et al., 2024a; Deeb and Roger, 2024; Doshi and Stickland, 2024; Lynch et al., 2024). Alternative approaches enable unlearning during inference (Wang et al., 2024b; Eldan and Russinovich, 2023; Ji et al., 2024; Thaker et al., 2024b; Liu et al., 2024a). In this work, we focus on fine-tuning methods, as they are widely adopted.

Representation Fine-tuning (ReFT). ReFT (Wu et al., 2024b) is a recently proposed parameter-efficient fine-tuning method. Unlike traditional fine-tuning approaches, which primarily adjust model weights, ReFT focuses on fine-tuning representations, leveraging the rich semantic information embedded in the representation space to influence subsequent generation. Building on the linear representation hypothesis (Park et al., 2023), which posits that concepts are encoded within linear subspaces of representations, ReFT learns low-rank linear transformations to refine representations. It achieves this by substituting the intermediate representations—i.e., the outputs of specific Transformer layers—at selected layers and tokens.

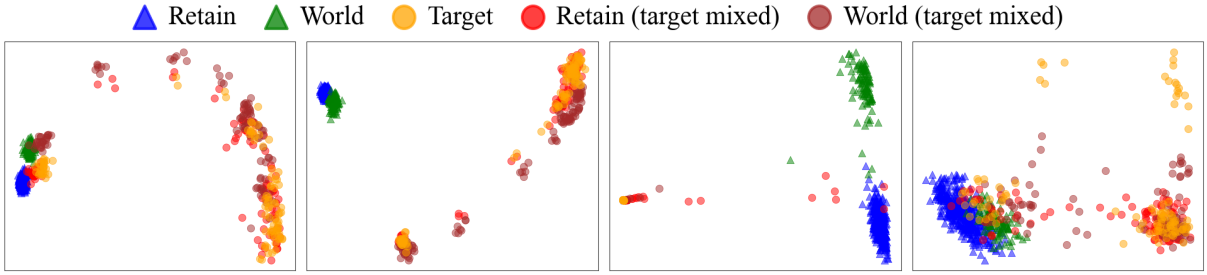
3 Preliminary studies

In this section, we first introduce the definitions about fine-tuning-based unlearning, and then conduct experiments to investigate their common properties.



(a) Llama 3.1 by GD (b) Llama 3.1 by NPO (c) Mistral v0.1 by GD (d) Mistral v0.1 by NPO

Figure 1: PCA visualizations of embeddings (both before and after unlearning) of target data, retaining data, and never-seen data. We apply 2-component PCA to project the embeddings into a 2D space and visualize the distributions. Each subfigure corresponds to a separate PCA projection for an unlearned model.



Not mixed: 0.483/0.785
Target data mixed: 0.010/0.000
(a) Llama 3.1 by GD

Not mixed: 0.551/0.851
Target data mixed: 0.157/0.050
(b) Llama 3.1 by NPO

Not mixed: 0.687/0.403
Target data mixed: 0.013/0.000
(c) Mistral v0.1 by GD

Not mixed: 0.842/0.494
Target data mixed: 0.225/0.030
(d) Mistral v0.1 by NPO

Figure 2: PCA visualization and the results of normal Q&A mixed and not mixed with target data. PCA follows the same operation in Figure 1. The ROUGE-L Recalls of retaining data/world fact are listed below each figure.

3.1 Fine-tuning-based unlearning

Given an LLM f and a target dataset $\mathcal{D}_\mathrm{t}$, the goal of an unlearning task is to get a model f_u that behaves as if it was never trained on $\mathcal{D}_\mathrm{t}$. Besides, f_u should also retain the model utility, i.e. the general text generation capabilities. To achieve this, various fine-tuning-based methods have been developed, such as GA-based and suppression-based methods.

In **GA-based methods**, the unlearning objective is usually formulated as the following:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ \mathbb{E}_{(x,y)\in\mathcal{D}_\mathrm{t}}\left[L_\mathrm{f}(y|x;\boldsymbol{\theta})\right] + \lambda\mathbb{E}_{(x,y)\in\mathcal{D}_\mathrm{r}}\left[L_\mathrm{r}(y|x;\boldsymbol{\theta})\right], \quad (1)$$

where $\mathcal{D}_\mathrm{r}$ is the retaining dataset to preserve the model utility, and (x, y) denotes an input-output pair. $\boldsymbol{\theta}$ represents the updated parameters, while L_f and L_r denote the forgetting and retaining loss functions, respectively, with λ balancing them. Typically, L_f is the negative training loss (i.e., applying Gradient Ascent) or a variant, while L_r corresponds to the training loss on $\mathcal{D}$r or a regularization term (e.g., the KL divergence between the f and f_u).

We introduce two GA-based methods. Gradient Difference (GD) (Liu et al., 2022) applies negative standard training loss on $\mathcal{D}_\mathrm{t}$ as L_f. Negative Preference Optimization (NPO)(Zhang et al., 2024a), derived from DPO (Rafailov et al., 2024), constrains divergence from the initial checkpoint to regulate strength of GA.

Suppression-based methods have a similar objective:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ \mathbb{E}_{(x,y)\in\mathcal{D}_\mathrm{t}}\left[L_\mathrm{s}(y,x,\boldsymbol{\theta})\right] + \lambda\mathbb{E}_{(x,y)\in\mathcal{D}_\mathrm{r}}\left[L_\mathrm{r}(y|x;\boldsymbol{\theta})\right],$$

L_s is the suppression term. In IDK (Maini et al., 2024), L_s encourages responses like "I don't know" for target data, while in RMU (Li et al., 2024), it pushes target data representations toward a random vector to disturb target data.

3.2 Findings of GA-based unlearning

In this subsection, we investigate the common property between GA-based and suppression-based methods. We find that GA-based methods cannot remove target data as expected. Instead, the GA-unlearned models distinguish the target data and pretend to be unaware. It is actually the same strategy as suppression-based methods. Experiments are conducted by exploring following questions.

(1) Does reversing the training loss truly negate the target data's influence?

If the GA-based methods could remove the influence of target data, it is expected that the unlearned models should behave the same between the target data and the data it has never encountered. To investigate this, we conduct an experiment to compare the model behaviors in these two data cases.

Settings. We use TOFU dataset, which contains synthetic Q&A pairs about non-existent writers and books. We split the dataset into three subsets: target data, retaining data and never-seen data. We first fine-tune LLMs to learn the knowledge from the target data and retaining data. Then we unlearn the target data by GD and NPO. In Figure 1, we plot the embeddings (both before and after unlearning) of target data, retaining data and never-seen data.

Results. In Figure 1, we observe that in embedding space, the unlearned models still recognize target data, and distinguish it with a special pattern. *Before* unlearning, the target data, retaining data, and never-seen data have similar embeddings, as all three sets are sampled from the same data distribution. In contrast, *after* unlearning, the target data follows a significantly different pattern, distributed far from the retaining and never-seen data. This suggests that the model does not truly remove the target data. Instead, they still recognize it and distinguish it by pushing it into a distinct region.

(2) Is unlearning performance associated with this distinct pattern?

To further explore the connection between unlearning and distinct patterns, we quantify the distinction and the unlearning effectiveness in Table 1.

Settings. We quantify the distinction using the degree of overlap between the embeddings of target and retained data, measured by Class-wise Separability Discriminant (CSD), i.e., the ratio of intra-class distance (samples within target and samples within retaining data) to inter-class distance (between target data and retaining data) (Ren et al., 2023; Klecka, 1980). Unlearning effectiveness is evaluated using ROUGE-L Recall, where a lower score on target data indicates better unlearning (as detailed in Section 5.1).

Observation. In Table 1, we observe that when the pattern is more distinct (i.e., lower CSD), the target data is more effectively unlearned (i.e., lower ROUGE-L Recall). For example, Mistral unlearned by GD has the lowest CSD and the lowest ROUGE-L Recall, while Llama unlearned by NPO has the highest CSD and the highest ROUGE-L Recall. This implies that better unlearning performance is likely to be associated with better distinction.

	Llama 3.1		Mistral v0.1	
	GD	NPO	GD	NPO
CSD	0.45	3.21	0.13	1.72
ROUGE-L Recall	0.016	0.197	0.001	0.127

Table 1: Unlearning effectiveness and distinction

(3) How do GA-based methods unlearn?

To analyze how the GA-unlearned models process the target data, we compare the model behaviors between target data and normal Q&A data (questions that should be correctly answered by unlearned models).

Settings. We inject target data into normal Q&A pairs to form the mixed data and compare the model's behaviors before and after the injection. We use retaining data and world fact Q&A pairs as normal Q&A data. For example, a mixed data instance is "*Where is Eiffel Tower? And who is the author of Watermelon on the Moon*?", where "*who is the author of . . .*" is an instance in target data. We plot the embeddings and calculate ROUGE-L Recall (higher score means more correct answers) in Figure 2.

Results. From Figure 2, we can see that the unlearned models actually treat target data as the unlearning signal. Specifically, before adding target data, the models correctly answer normal questions, achieving a high ROUGE-L Recall. However, once mixed with target data, the embeddings of normal data is dominated by target data (which is pulled toward the distinct area of target data). Consequently, the model's ability to answer normal questions deteriorates (lower ROUGE-L). This implies that, instead of removing the target data, GA-unlearned models treat it as a suppression signal.

In summary, our preliminary studies reveal that GA-based unlearning methods do not erase the target data as expected. Instead, the models still recognize it and distinguish it in the embeddings. Unlearning performance is likely to be associated with the distinction. When target data appears in the prompt, the model suppresses related generations—essentially employing the same strategy as suppression-based methods.

4 Method

In this section, we first present the design of GRUN and its training procedure. Lastly, we discuss how to extend GRUN to sequential unlearning, where multiple unlearning requests occur over time.

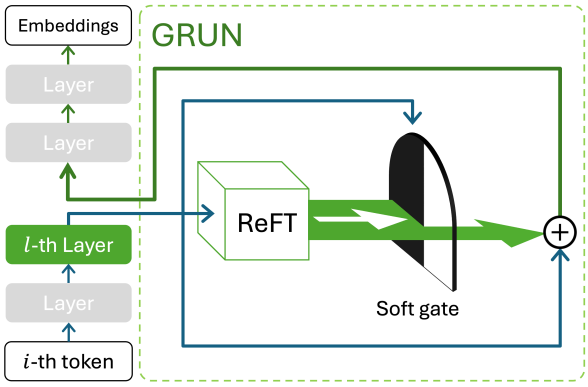


Figure 3: An overall of the framework of GRUN.

4.1 GRUN

The observation in our preliminary study suggests that the mechanism of both GA-based and suppression-based methods is to distinguish the target data. Based on this, we proposed the ReFT-based Gated Representation UNlearning method to explicitly take the advantage of this finding.

An overview of GRUN is in Figure 3. GRUN consists of two plug-and-play components explicitly for distinguishing and suppression: a soft gate function to distinguish target data, and a ReFT module to suppress target-data-related generation. We first explain the elements of ReFT below.

ReFT. As shown in Section 2, ReFT modifies a model by freezing its parameters while fine-tuning the intermediate representations of some layers. Specifically, it applies the following transformation to update the d-dimensional representation $\boldsymbol{h}_i^{(l)}$ of the i-th token at layer l:

$$\Phi_{\text{ReFT}}(\boldsymbol{h}_i^{(l)}) = \boldsymbol{h}_i^{(l)} + \phi(\boldsymbol{h}_i^{(l)}),$$

where $\phi(\boldsymbol{h}_i^{(l)})$ is a trainable low-rank linear transformation defined as

$$\phi(\boldsymbol{h}_i^{(l)}) = \mathbf{R}^\top(\mathbf{W}\boldsymbol{h}_i^{(l)} + \boldsymbol{b} - \mathbf{R}\boldsymbol{h}_i^{(l)}), \quad (2)$$

where $\mathbf{R} \in \mathbb{R}^{r\times d}$, $W \in \mathbb{R}^{r\times d}$ and $b \in \mathbb{R}^r$ are trainable parameters, with $r \ll d$. Intuitively, the term $\mathbf{W}\boldsymbol{h}_i^{(l)} + \boldsymbol{b}$ represents the target representation we aim to shift towards, while $\phi(\boldsymbol{h}_i^{(l)})$ is the directional adjustment from $\boldsymbol{h}_i^{(l)}$ to the target representation in the space defined by $\mathbf{R}$. By replacing the original representation $\boldsymbol{h}_i^{(l)}$ with the new representation $\Phi_{\text{ReFT}}(\boldsymbol{h}_i^{(l)})$, ReFT modifies the embeddings of the input, thereby influencing the subsequent generation. A figure of ReFT is in Appendix A.

GRUN. On top of ReFT, we define GRUN as:

$$\Phi_{\text{GRUN}}(\boldsymbol{h}_i^{(l)}) = \boldsymbol{h}_i^{(l)} + g(\boldsymbol{h}_i^{(l)})\phi(\boldsymbol{h}_i^{(l)}), \quad (3)$$

where g is the gate function. More specifically, the soft gate g is a single-output regression model (linear regression or Multi-Layer Perceptron neural network) with a softmax function following the output. Thus, the output value of g is in the range of (0,1). As shown in Figure 3, when the input representation $\boldsymbol{h}_i^{(l)}$ is related to the target data, $g(\boldsymbol{h}_i^{(l)})$ is closed to 1 which starts the low-rank transform for unlearning. In contrast, if the input is not about target data, then $g(\boldsymbol{h}_i^{(l)})$ is closed to 0 which passes limited changes on the representation.

While GRUN can be used in any token position and any Transformer layer, the configuration in our work is as follows:

(1) The last token of input usually contains all the semantic information of the input and has a significant impact on the generation. Thus, we use GRUN at the last token position of input. (2) To improve effectiveness, we use GRUN for multiple layers in a model instead of a single layer. Since the later layers capture higher-level semantics than previous layers which are beneficial for the distinguishing of gate function, we choose to use GRUN for later layers (Peng et al., 2018; Jin et al., 2025). To reduce the mutual influence (as discussed in Appendix B.1), we choose interval layers instead of successive layers. Specifically, for the LLMs studied in the following work, the layers are: the last layer, the last 7th layer and the last 12th layer.

4.2 Training objective

Our method is a unified method that can be adapted to different fine-tuning based unlearning loss such as GA (Yao et al., 2023), GD (Liu et al., 2022), NPO (Zhang et al., 2024a), IDK (Maini et al., 2024), RMU (Li et al., 2024) and other fine-tuning based methods. In other words, GRUN can be also seen as a new fine-tuning method that is tailored for the LLM unlearning task.

The training objective is represented as follows:

$$\begin{aligned} L =& L_{\text{u}} + L_{\text{G}} \quad (4)\\ =& L_{\text{u}} + \mathbb{E}_{(x,y,\hat{y})\in\mathcal{D}_{\text{t}}\cup\mathcal{D}_{\text{r}}}\mathbb{E}_{i,l}L_{\text{CE}}\left(g(\boldsymbol{h}_i^{(l)}), \hat{y}\right), \end{aligned}$$

where L_{u} is an unlearning loss which can be GA-based or suppression-based loss, $\hat{y}$ is the label to indicate target data ($\hat{y} = 1$) and retain data ($\hat{y} = 0$), and L_{G} is the cross-entropy loss for the output

of gate function. The unlearning loss L_{u} is used to ensure the unlearning purpose. The term L_{G} fine-tunes the gate function to open (closer to 1) for target data more and close (closer to 0) for the other data. This training objective distinguishes the target data for unlearning and keeps the model utility by minimizing its impact on the normal input.

4.3 Sequential unlearning

In real-world scenarios, the unlearning requests typically arise sequentially over time. To process this sequential unlearning, previous methods have to re-train the whole set or fine-tune multiple rounds which would largely reduce the model utility due to the accumulated parameter distortion (Shi et al., 2024b). In contrast, in GRUN, we mitigate this by using an independent ReFT for each unlearning request and combine them together. Specifically, if we have $M-1$ unlearning requests finished and get the new M-th request, we can fine-tune a separate gate for the new coming target set and combine multiple GRUNs by

$$\Phi^{M}_{\text{GRUN}}(\boldsymbol{h}_i^{(l)}) = \boldsymbol{h}_i^{(l)} + c \sum_{j=1}^{M} g_j(\boldsymbol{h}_i^{(l)})\phi(\boldsymbol{h}_i^{(l)}),$$

where c is the coefficient to balance the strength. Each gate g_j is fine-tuned independently on a requested target dataset $\mathcal{D}_{\text{t},j}$ and then combined. The coefficient c reduces as the increasing of M (the details to determine c is in Appendix B.2). In this way, we can mostly preserve the model utility and save the training efforts.

5 Experiment

In this section, we first conduct the experiments across different models and datasets in Section 5.2. Then we test the performance under different scenarios including sequential unlearning and attacks in Sec. 5.3, and conduct ablations studies in Section 5.4 and Appendix 5.

5.1 Experimental settings

Models, baselines and datasets. We use Llama 3.1 (8B) (Dubey et al., 2024) and Mistral v0.1 (7B) (Jiang et al., 2023). We experiment on two datasets TOFU (unlearn fine-tuning knowledge) and WMDP (unlearn pre-training knowledge). Following the original settings (Maini et al., 2024), we use GD, NPO, and IDK as baselines (using both vanilla and LoRA fine-tuning) in TOFU. Following Li et al. (2024), we use RMU as the baseline in WMDP. GD and NPO are GA-based, while IDK and RMU are suppression-based.

Metrics. For TOFU, we use ROUGE-L Recall and Probability following Maini et al. (2024). ROUGE-L Recall assesses correctness of the output text, while Probability reflects the likelihood of generating correct responses (Appendix C.1 for details). WMDP consists of multi-choice Q&A, therefore, we use the accuracy as the metric to access whether the model can correctly answer the questions following Li et al. (2024). For all the three metrics, lower scores on target data indicate better erasing, while higher scores on normal data indicates better utility. Time cost is measured in GPU hours (number of GPUs $\times$ training hours).

Implementation details. For baselines, GD, NPO and IDK follow Fan et al. (2024), while RMU follows Li et al. (2024). For GRUN, adapted NPO, IDK, and RMU are trained for fixed epochs, while GD uses early stop when L_{f} in Eq. (1) exceeds the threshold. We use linear regression as gate for Llama and 3-layer MLP for Mistral. Both LoRA and GRUN use rank of 4. All other details are in Appendix C.2.

5.2 Main results

In this subsection, we present the results of TOFU and WMDP and compare the time cost of GRUN with vanilla fine-tuning and LoRA. The unlearning assessment consists of two aspects: (1) the extent to which the target data can be removed/unlearned (*unlearning effectiveness*), and (2) the preservation of model *utility*.

TOFU. To evaluate on TOFU, we compare unlearning effectiveness, utility, and time cost against three baselines on two LLMs in Table 2. The LLMs are first fine-tuned on TOFU's synthetic dataset, after which a portion of the dataset is designated as the target data for unlearning, while the remaining synthetic data serves as the retaining data for utility. Utility is assessed on three sets of data: retained data, Q&A about real authors, and Q&A about world facts, with the overall utility being their average. From Table 2, our method consistently outperforms the baselines of vanilla fine-tuning.

Specifically, for *GD*, GRUN has similar unlearning effectiveness as the vanilla baseline, while significantly improving the utility, particularly in ROUGE-L Recall, where it achieves an increase of around 20% for both Llama 3.1 and Mistral v0.1.

For *NPO*, our method substantially enhances its unlearning effectiveness while also achieving even

L_u	LLM	p_{tgt}	Method	p_{size}	Hours	ROUGE-L Recall		Prob.	
						Unlearn↓	Utility(Retain/Fact/World)↑	Unlearn↓	Utility(Retain/Fact/World)↑
	Llama	5%	Clean	N/A	N/A	0.991	0.939 (0.992/0.939/0.890)	0.995	0.566 (0.993/0.448/0.485)
		10%				0.992		0.995	
	Mistral	5%	Clean	N/A	N/A	0.990	0.710 (0.994/0.515/0.622)	0.994	0.610 (0.995/0.401/0.433)
		10%				0.988		0.990	
GD	Llama	5%	Vanilla	100%	3.19	0.005	0.703 (0.493/0.854/0.762)	0.000	0.605 (0.575/0.622/0.619)
			GRUN	**0.001%**	**0.02**	**0.002**	**0.843** (0.888/0.843/0.798)	**0.000**	0.584 (0.874/0.432/0.446)
		10%	Vanilla	100%	6.33	0.005	0.695 (0.483/0.818/0.785)	0.000	0.554 (0.654/0.496/0.513)
			GRUN	**0.001%**	**0.02**	0.016	**0.832** (0.906/0.729/0.862)	0.006	**0.592** (0.912/0.402/0.462)
	Mistral	5%	Vanilla	100%	3.01	0.004	0.568 (0.742/0.360/0.601)	0.000	0.581 (0.829/0.448/0.466)
			GRUN	**0.045%**	**0.06**	**0.000**	**0.660** (0.956/0.485/0.539)	**0.000**	**0.588** (0.955/0.417/0.391)
		10%	Vanilla	100%	6.07	0.001	0.396 (0.687/0.099/0.403)	0.000	0.558 (0.830/0.358/0.485)
			GRUN	**0.045%**	**0.18**	**0.000**	**0.595** (0.891/0.390/0.504)	**0.000**	0.545 (0.886/0.354/0.395)
NPO	Llama	5%	Vanilla	100%	3.96	0.201	0.751 (0.616/0.756/0.883)	0.016	0.645 (0.766/0.546/0.623)
			GRUN	**0.001%**	**0.19**	**0.020**	**0.886** (0.973/0.857/0.828)	**0.000**	0.634 (0.977/0.447/0.477)
		10%	Vanilla	100%	7.93	0.197	0.738 (0.551/0.811/0.851)	0.025	0.599 (0.730/0.465/0.602)
			GRUN	**0.001%**	**0.38**	**0.029**	**0.862** (0.928/0.849/0.811)	**0.000**	**0.599** (0.911/0.441/0.446)
	Mistral	5%	Vanilla	100%	3.50	0.163	0.530 (0.820/0.256/0.514)	0.030	0.558 (0.912/0.364/0.399)
			GRUN	**0.045%**	**0.16**	**0.000**	**0.675** (0.984/0.485/0.555)	**0.000**	**0.596** (0.980/0.394/0.414)
		10%	Vanilla	100%	6.99	0.127	0.542 (0.842/0.290/0.494)	0.024	0.567 (0.923/0.360/0.419)
			GRUN	**0.045%**	**0.34**	**0.000**	**0.637** (0.893/0.445/0.573)	**0.000**	0.531 (0.890/0.342/0.362)
IDK	Llama	5%	Vanilla	100%	1.65	0.023	0.672 (0.578/0.627/0.812)	0.468	0.623 (0.871/0.479/0.520)
			GRUN	**0.001%**	**0.08**	**0.021**	**0.905** (0.980/0.882/0.853)	**0.261**	**0.625** (0.984/0.434/0.458)
		10%	Vanilla	100%	3.33	0.023	0.547 (0.570/0.353/0.718)	0.532	0.614 (0.871/0.459/0.512)
			GRUN	**0.001%**	**0.18**	**0.023**	**0.865** (0.892/0.879/0.823)	**0.291**	0.605 (0.938/0.435/0.441)
	Mistral	5%	Vanilla	100%	1.53	0.023	0.435 (0.785/0.122/0.399)	0.533	0.574 (0.962/0.366/0.395)
			GRUN	**0.045%**	**0.09**	**0.022**	**0.683** (0.975/0.480/0.593)	0.570	**0.606** (0.987/0.401/0.430)
		10%	Vanilla	100%	3.07	0.023	0.489 (0.856/0.145/0.466)	0.657	0.595 (0.975/0.392/0.417)
			GRUN	**0.045%**	**0.20**	0.040	**0.605** (0.914/0.430/0.469)	**0.490**	0.577 (0.953/0.394/0.386)

Table 2: Results of TOFU. p_{tgt} represents the proportion of target data within the entire synthetic dataset. p_{size} is the percentage of fine-tuned parameters relative to the entire LLM. "Unlearn" refers to the unlearning effectiveness, and "Clean" refers to the model before unlearning. The improved performance is highlighted in **bold**.

higher utility. For example, on Llama, our approach reduces NPO's ROUGE-L Recall on the target data from approximately 0.2 to 0.02 while increasing utility by around 17.5%.

As for *IDK*, which is suppression-based, its vanilla version has a more severe impact on the utility of author-related Q&A (both synthetic and real) than GA-based methods. However, our method significantly improves utility performance, increasing ROUGE-L Recall by more than 25% in most cases.

From Table 2, we also observe that GRUN is more efficient, requiring fewer parameters and lower training costs. We defer the discussion to following Table 4 for LoRA experiments.

WMDP. Table 3 presents the results of removing pre-training knowledge in WMDP. WMDP evaluates unlearning by erasing harmful biological and cyber knowledge while assessing utility using the benign Q&A dataset MMLU (Hendrycks et al., 2020). WMDP uses a 4-choice Q&A to measure the knowledge. We adjust the unlearning strength to maintain similar utility between vanilla RMU and GRUN, and only compare the unlearning effectiveness. In Table 3, our approach significantly improves performance on Llama 3.1 and maintains a random-guessing accuracy on Mistral v0.1.

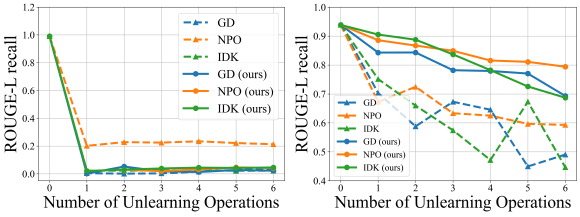
RMU	Llama 3.1		Mistral v0.1	
	Bio/Cyber↓	MMLU↑	Bio/Cyber↓	MMLU↑
Before	0.696/0.418	0.611	0.668/0.437	0.581
Vanilla	0.494/0.337	0.581	**0.256/0.252**	0.529
GRUN	**0.372/0.293**	0.577	0.293/0.278	0.535

Table 3: Unlearning results on WMDP

			p_{size}	Hours	ROUGE-L		Prob.	
				unlearn	utility	unlearn	utility	
Llama 3.1	GD	LoRA	0.130%	1.27	0.375	0.623	0.059	0.067
		GRUN	**0.001%**	**0.02**	**0.000**	**0.840**	**0.000**	**0.582**
	NPO	LoRA	0.130%	0.77	0.255	0.886	0.103	0.315
		GRUN	**0.001%**	**0.08**	**0.020**	**0.896**	**0.000**	**0.634**
	IDK	LoRA	0.130%	1.33	0.054	0.782	0.849	0.346
		GRUN	**0.001%**	**0.19**	**0.021**	**0.915**	**0.262**	**0.625**

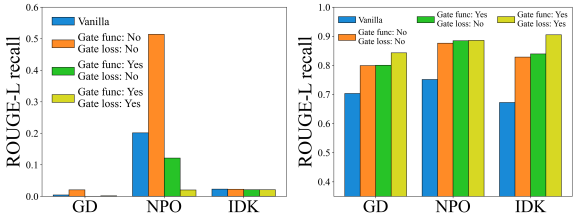
Table 4: Comparison with LoRA

LoRA. We compare GRUN with LoRA to further



(a) Unlearning effectiveness (b) Model utility

Figure 4: Sequential unlearning



(a) Unlearning effectiveness (b) Model utility

Figure 5: Contributions of each components

Effectiveness	Paraphrase		Quantization	
	Llama	Mistral	Llama	Mistral
GD (GRUN)	0.006	0.005	0.002	0.000
NPO (GRUN)	0.019	0.000	0.021	0.000
IDK (GRUN)	0.044	0.040	0.038	0.034

Table 5: Unlearning effectiveness under attacks

demonstrate its superiority in efficiency. As shown in Table 4, our method requires fewer parameters while achieving better performance across all unlearning and utility metrics, regardless of the model or fine-tuning loss. Additionally, GRUN reduces training time by 95% compared to vanilla training (Table 2) and by 85% compared to LoRA. This efficiency gain is attributed to two key factors:

- *Fewer parameters to update.* GRUN updates less than 0.05% (even 0.001% for Llama) of the parameters compared to the full LLM.
- *A significantly shorter gradient backpropagation path.* GRUN is applied only to the last few layers, eliminating the computational cost of backpropagating gradients through the earlier layers. (LoRA updates fewer parameters, but has to backpropagate the entire network.)

5.3 Different unlearning scenarios

In this subsection, we evaluate GRUN's performance under sequential unlearning and assess its robustness against two attacks—prompt paraphrasing and model quantization—to validate its effectiveness across various unlearning scenarios.

Sequential unlearning. In Figure 4, we first fine-tune the models with all the synthetic data of TOFU, and then simulate sequential unlearning by issuing six unlearning requests, each targeting a different forget set containing 5% synthetic data. As shown in Figure4a, the unlearning effectiveness remains consistent across both baselines and our method. However, in Figure 4b, our approach significantly outperforms the baselines in utility when multiple requests are processed.

Robustness. In Table. 5, we evaluate the robustness of GRUN by attacking the unlearned model to recover the removed knowledge through prompt paraphrasing and model quantization. We use GPT-4 to paraphrase the questions to bypass GRUN's distinguishing mechanism. Our method remains stable, preserving the original unlearning effectiveness. Zhang et al. (2024b) reports that quantization may negate unlearning; however, our approach effectively recognizes and removes quantized representations with no loss in effectiveness.

5.4 Ablation study

In this subsection, we conduct ablation studies to analyze the effects of each component of GRUN, i.e., ReFT, the soft gate, and the gate loss (L_{G}).

We compare vanilla fine-tuning along with three variants of GRUN to evaluate the contribution of each component: (1) ReFT-only (without the gate or L_{G}), (2) GRUN without L_{G} (maintaining the same structure as GRUN but trained solely with L_{u}), and (3) the complete GRUN.

ReFT-only. In Figure 5, switching from vanilla fine-tuning to ReFT-only increases utility but reduces unlearning effectiveness. This suggests that ReFT enhances utility by freezing model parameters as expected but has limited capability in distinguishing target data due to its simple structure.

GRUN without L_{G}. Adding the gate function (without L_{G}), improves unlearning effectiveness, particularly for NPO. This indicates that even in the absence of L_{G}, the gate function can automatically aid in distinguishing target data during optimization. (More empirical analysis in Appendix D.)

The complete GRUN. The complete GRUN model further enhances both unlearning effectiveness and utility. This demonstrates that explicitly guiding GRUN with L_{G} fundamentally strengthens fine-tuning-based methods.

6 Conclusions

Unlearning aims to remove copyrighted and privacy-sensitive data from LLMs, but often degrades model utility. We propose GRUN, a general framework to enhances fine-tuning-based unlearning. GRUN leverages the shared mechanism between GA-based and suppression-based methods. It uses a soft gate function for distinguishing and a ReFT-based suppression module to adjust representations. GRUN improves both unlearning effectiveness and utility, and enables efficient unlearning.

Limitations

Although our method can largely enhance the performance of unlearning, our method still cannot achieve truly unlearning. We start from the current methods to discuss how far from the real goal in this work. Current LLMs are vast and complex, making it difficult to precisely locate and remove specific knowledge traces. Future research is needed to develop more robust and theoretically grounded approaches to precisely track, isolate, and eliminate specific knowledge without compromising overall model capabilities. We hope this work can inpsire further exploration into this pressing issue.

Acknowledgment

Jie Ren and Hui Liu are supported by the National Science Foundation (NSF) under grant numbers CNS2321416, IIS2212032, IIS2212144, IOS2107215, DUE2234015, CNS2246050, DRL2405483 and IOS2035472, US Department of Commerce, Gates Foundation, the Michigan Department of Agriculture and Rural Development, Amazon, Meta, and SNAP.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Zhiqi Bu, Xiaomeng Jin, Bhanukiran Vinzamuri, Anil Ramakrishna, Kai-Wei Chang, Volkan Cevher, and Mingyi Hong. 2024. Unlearning as multi-task optimization: A normalized gradient difference approach with an adaptive learning rate. *arXiv preprint arXiv:2410.22086*.

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

Sungmin Cha, Sungjun Cho, Dasol Hwang, and Moontae Lee. 2024. Towards robust and cost-efficient knowledge unlearning for large language models. *arXiv preprint arXiv:2408.06621*.

Aghyad Deeb and Fabien Roger. 2024. Do unlearning methods remove information from language model weights? *arXiv preprint arXiv:2410.08827*.

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

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

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

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. 2024. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *arXiv preprint arXiv:2410.07163*.

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

Philipp Hacker, Andreas Engel, and Marco Mauer. 2023. Regulating chatgpt and other large generative ai models. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1112–1123.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

James Y Huang, Wenxuan Zhou, Fei Wang, Fred Morstatter, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2024. Offset unlearning for large language models. *arXiv preprint arXiv:2404.11045*.

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

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 *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408.

Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana Rao Kompella, Sijia Liu, and Shiyu Chang. 2024. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *arXiv preprint arXiv:2406.08607*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, et al. 2025. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 558–573.

William R Klecka. 1980. *Discriminant analysis*. Sage.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. 2024. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. 2024. The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *Forty-first International Conference on Machine Learning*.

Yuantong Li, Chi-Hua Wang, and Guang Cheng. 2021. Online forgetting process for linear regression models. In *International Conference on Artificial Intelligence and Statistics*, pages 217–225. PMLR.

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

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

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. 2024b. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787*.

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

Nicola Lucchi. 2024. Chatgpt: a case study on copyright challenges for generative artificial intelligence systems. *European Journal of Risk Regulation*, 15(3):602–624.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

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

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. 2024. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few-shot unlearners. In *Forty-first International Conference on Machine Learning*.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 world wide web conference*, pages 1063–1072.

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

Jie Ren, Han Xu, Pengfei He, Yingqian Cui, Shenglai Zeng, Jiankun Zhang, Hongzhi Wen, Jiayuan Ding, Pei Huang, Lingjuan Lyu, et al. 2024. Copyright protection in generative ai: A technical perspective. *arXiv preprint arXiv:2402.02333*.

Jie Ren, Han Xu, Yuxuan Wan, Xingjun Ma, Lichao Sun, and Jiliang Tang. 2023. Transferable unlearnable examples. In *The Eleventh International Conference on Learning Representations*.

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

Shaojie Shi, Xiaoyu Tan, Xihe Qiu, Chao Qu, Kexin Nie, Yuan Cheng, Wei Chu, Xu Yinghui, and Yuan Qi. 2024a. Ulmr: Unlearning large language models via negative response and model parameter average. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 755–762.

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

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

Yash Sinha, Murari Mandal, and Mohan Kankanhalli. 2024. Unstar: Unlearning with self-taught anti-sample reasoning for llms. *arXiv preprint arXiv:2410.17050*.

Pratiksha Thaker, Shengyuan Hu, Neil Kale, Yash Maurya, Zhiwei Steven Wu, and Virginia Smith. 2024a. Position: Llm unlearning benchmarks are weak measures of progress. *arXiv preprint arXiv:2410.02879*.

Pratiksha Thaker, Yash Maurya, Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. 2024b. Guardrail baselines for unlearning in llms. *arXiv preprint arXiv:2403.03329*.

Bozhong Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Mengru Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. 2024. To forget or not? towards practical knowledge unlearning for large language models. *arXiv preprint arXiv:2407.01920*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Akshaj Kumar Veldanda, Shi-Xiong Zhang, Anirban Das, Supriyo Chakraborty, Stephen Rawls, Sambit Sahu, and Milind Naphade. 2024. Llm surgery: Efficient knowledge unlearning and editing in large language models. *arXiv preprint arXiv:2409.13054*.

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

Qizhou Wang, Bo Han, Puning Yang, Jianing Zhu, Tongliang Liu, and Masashi Sugiyama. 2024a. Unlearning with control: Assessing real-world utility for large language model unlearning. *arXiv preprint arXiv:2406.09179*.

Shang Wang, Tianqing Zhu, Dayong Ye, and Wanlei Zhou. 2024b. When machine unlearning meets retrieval-augmented generation (rag): Keep secret or forget knowledge? *arXiv preprint arXiv:2410.15267*.

Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Parag Shah, Yujia Bao, Yang Liu, and Wei Wei. 2024c. Llm unlearning via loss adjustment with only forget data. *arXiv preprint arXiv:2410.11143*.

Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. 2021. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*.

Ruihan Wu, Chhavi Yadav, Russ Salakhutdinov, and Kamalika Chaudhuri. 2024a. Evaluating deep unlearning in large language models. *arXiv preprint arXiv:2410.15153*.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2024b. ReFT: Representation finetuning for language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the catastrophic forgetting in multimodal large language models. *arXiv preprint arXiv:2309.10313*.

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

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. 2024b. Catastrophic failure of llm unlearning via quantization. *arXiv preprint arXiv:2410.16454*.

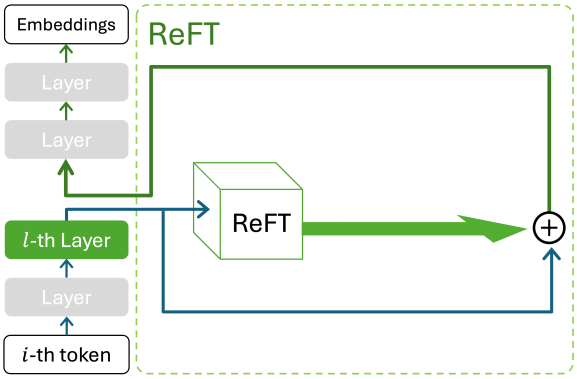


Figure 6: An overall of the framework ReFT.

L_G	L_u	Gate 1 ($l = 20$) target ↑	Gate 1 ($l = 20$) retain ↓	Gate 2 ($l = 25$) target ↑	Gate 2 ($l = 25$) retain ↓	Gate 3 ($l = 31$) target ↑	Gate 3 ($l = 31$) retain ↓
No	GD	0.00	0.00	0.99	0.08	1.00	0.05
	NPO	1.00	1.00	1.00	1.00	1.00	1.00
	IDK	1.00	1.00	0.00	0.00	0.00	0.00
Yes	GD	0.93	0.24	1.00	0.03	0.92	0.02
	NPO	0.99	0.09	1.00	0.02	1.00	0.02
	IDK	0.99	0.09	1.00	0.02	1.00	0.01

Table 6: Outputs of gate functions. $l = 20, 25, 31$ represents the last 12th, last 7th and last layer respectively. The arrow ↑ (or ↓) means the output is expected to be close to 1 (or 0).

A ReFT

The figure of ReFT is shown in Figure 6.

B Hyper-parameters

B.1 Choosing layers for GRUN

We find that when the layers are too close, it is possible to influence the each other's training. For example, when we use the last two layers for GRUN, the unlearning performance of Llama increases to 0.4 for GD. Thus, we use interval layers.

B.2 The coefficient c for sequential unlearning

In our experiments, we tune the hyper-parameter c to get the best utility while maintaining the unlearning. This is reasonable since the LLM builder have the target data which can be used to search the best hyper-parameters.

C Experimental settings.

C.1 Metrics

For Probability, TOFU uses the normalized likelihood for target and retaining data. For real authors and world facts, we follow their settings use the probability between correct answer and paraphrased answer (wrong answers). Please refer the details to Maini et al. (2024).

C.2 Other implementation details.

GRUN is trained for 40 epochs on NPO and IDK. All the learning rates are 1e-5. The time cost is tested on A6000 GPUs.

D Additional experiments

To further examine the different behaviors of the gate function with and without L_G, we present the gate function outputs for target and retaining data in Table 6. With L_G, the gate function behaves as expected—opening for target data while closing for retaining data. Even in the absence of explicit guidance from L_G, the gate can still differentiate effectively, as seen in Gate 2 and Gate 3 of GD. For IDK, the gate function helps identify the optimal layer for ReFT and adjusts by closing redundant layers. A special case arises with NPO when L_G is absent: all gates remain open for both target and retaining data. Although this structure appears similar to ReFT-only, it has significantly enhanced unlearning effectiveness compared with ReFT-only. We conjecture that the soft gate influences the optimization process. In the case of ReFT-only, retaining data may compete with target data due to their reversed losses. For GRUN without L_G, the gate may prioritize forgetting data early in training, as the loss on retaining data has limited room to decrease—having already converged before unlearning. This hypothesis is supported by the observation that, within the first 10 steps, the forgetting loss of GRUN without L_G is lower than that of ReFT-only.

E Failure cases

GRUN can occasionally harm model utility. For instance, in the TOFU benchmark, unlearned models sometimes fail on questions involving real authors, such as "Who is the author of the play 'Romeo and Juliet'?". GRUN is trained on the retaining data of TOFU, but not real authors. Thus, GRUN is good at distinguishing retaining data, but may not generalize well to other data like real authors. This highlights the importance of the diversity of retaining data.

F Details about capacity in sequential unlearning

GRUN demonstrates strong capacity in sequential unlearning, maintaining utility (measured by

ROUGE recall) above 0.5 even after 11 unlearning rounds. In contrast, the baseline NPO fails earlier, with utility dropping below 0.5 after just 6 unlearning steps.