Adaptive Localization of Knowledge Negation for Continual LLM Unlearning

Abudukelimu Wuerkaixi [1] **Qizhou Wang** [2] **Sen Cui** [1] **Wutong Xu** [1] **Bo Han** [2 3] **Gang Niu** [3]
Masashi Sugiyama [3 4] **Changshui Zhang** [1]

Abstract

With the growing deployment of large language models (LLMs) across diverse domains, concerns regarding their safety have grown substantially. LLM unlearning has emerged as a pivotal approach to removing harmful or unlawful content while maintaining utility. Despite increasing interest, the challenges of continual unlearning, which is common in real-world scenarios, remain underexplored. Successive unlearning tasks often lead to intensified utility degradation. To effectively unlearn targeted knowledge while preserving LLM utility, it is essential to minimize changes in model parameters by selectively updating those linked to the target knowledge, thereby ensuring other knowledge remains unaffected. Building on the task vector framework, we propose a new method named ALKN(Adaptive Localization of Knowledge Negation), which uses dynamic masking to sparsify training gradients and adaptively adjusts unlearning intensity based on inter-task relationships. Comprehensive experiments across three well-established LLM unlearning datasets demonstrate that our approach consistently outperforms baseline methods in both unlearning effectiveness and utility retention under continual unlearning settings.[1]

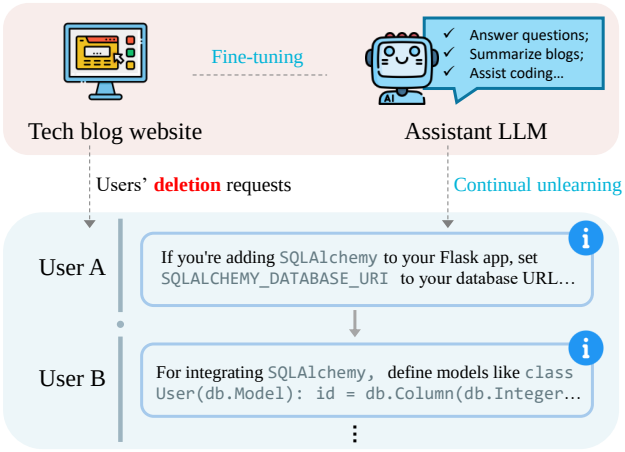


Figure 1. An illustrative example of the LLM continual unlearning scenario. An LLM is integrated into a technical blog website, where some users occasionally close their accounts and request the deletion of their blog contents.

1. Introduction

Large Language Models (LLMs) exhibit exceptional capabilities in processing general knowledge, positioning themselves as a promising framework on the path towards artificial general intelligence (Rozière et al., 2023; Wang et al., 2024b; Brown et al., 2020; Zhao et al., 2023). However, reliance on extensive web data introduces risks of memorizing sensitive information, including private data, copyrighted material, and harmful content, hindering lawful deployments of LLMs (Ji et al., 2024; Wang et al., 2025a). These factors continuously spur growing interest in research on LLM unlearning, which aims to remove specified sensitive information from LLMs while retaining their general knowledge and capabilities, i.e., preserving model utility. It focuses on ensuring that the model cannot reproduce or accurately respond to target data associated with harmful outcomes or legal issues (Liu et al., 2024a).

Although LLM unlearning has received much interest, an important problem of LLM continual unlearning remains underexplored. In real-world applications, unlearning requests from users or regulators often arrive continuously, requiring models to unlearn multiple pieces of data over time. For example, we consider a website equipped with a dedicated LLM that has been fine-tuned using user data from

[1]Institute for Artificial Intelligence, Tsinghua University (THUAI); Beijing National Research Center for Information Science and Technology (BNRist); Department of Automation, Tsinghua University, Beijing, P.R.China [2]TMLR Group, Department of Computer Science, Hong Kong Baptist University [3]RIKEN [4]The University of Tokyo. Correspondence to: Sen Cui <cuis@mail.tsinghua.edu.cn>, Changshui Zhang <zcs@mail.tsinghua.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

[1]Code is available at: https://github.com/zaocan666/ALKN

the platform, as shown in Figure 1. Periodically, users may request to delete their accounts and have their personal data removed, or regulators may report sensitive information in the LLM that needs to be removed. In these cases, the LLM needs to continuously perform unlearning tasks to ensure compliance with the requests from users and regulators.

Maintaining model utility represents a critical challenge in LLM unlearning, which is particularly pronounced in the continual unlearning scenario (Shi et al., 2024b). Representative baseline approaches to LLM unlearning use gradient ascent optimization (GA) and its variants (Han et al., 2020; Yao et al., 2023), inherently affecting the overall utility of the model due to the extreme modification of model parameters. The issue of utility deterioration is further magnified in the context of continual unlearning, which can be attributed to two factors. (I) Each unlearning task contributes to a decline in the model utility of the LLM. And such decline accumulates progressively with an increasing number of unlearning tasks. We refer to this phenomenon as *accumulative decline*. (II) After completing preceding unlearning tasks, the memory retention of the LLM regarding data from subsequent tasks may be disrupted, making them "partially forgotten" even before they are explicitly unlearned. Employing a fixed unlearning intensity without adjustment for these partially forgotten tasks can easily result in over-unlearning, further exacerbating the decline in model utility, which we refer to as *cascading degradation*. While accumulative decline stems from the progressive loss of utility over multiple unlearning tasks, cascading degradation arises from the unintended over-unlearning of partially forgotten data, amplifying the utility loss in subsequent tasks. These two issues combined exacerbate the utility degradation problem, making continual unlearning in LLMs a significantly more challenging scenario compared to one-time unlearning.

In this paper, we propose ALKN (Adaptive Localization of Knowledge Negation) to mitigate the problem of utility degradation in LLM continual unlearning. ALKN is built on the task vector framework, which first fine-tunes a model to reinforce knowledge from unlearning data and then negates the target knowledge by subtracting the parameter offsets from the original model. (Ilharco et al., 2023). Task vectors are trained with a lower-bounded loss, which inherently reduces the risk of excessive unlearning compared to GA. To further address the accumulative decline and cascading degradation issues in continual unlearning, we propose three key modules incorporated into the fine-tuning process of task vectors. (I) Entropic soft label loss strategically adjusts unlearning intensity via adaptive training objectives, reducing cascading degradation. (II) Dynamic gradient sparsity facilitates selective fine-tuning of different parameter sets for distinct tasks, minimizing accumulative decline. (III) Adaptive parameter modulation enables parameter-specific learning rates to further mitigate utility loss. In summary, the proposed modules, along with the task vector method, effectively address the catastrophic utility decline commonly observed in LLM continual unlearning by dynamically localizing model parameters and adaptively refining gradients.

We conducted comprehensive experiments to validate the effectiveness of our proposed method. To objectively assess the utility of models, we introduced Training Data Evaluation Corpus (TRAVIS) generated using the membership inference attacks (MIA) method (Shi et al., 2024a), designed to reconstruct knowledge in pretraining data of LLMs across a diverse range of topics. Testing on diverse training data provides a more comprehensive assessment of the overall utility of LLMs and also highlights precise utility decline relative to their pre-unlearning performance (Thaker et al., 2024). Experiments are performed across multiple models and datasets, demonstrating that ALKN effectively retains the general utility of the model while achieving effective unlearning of sequential tasks. After sufficient unlearning, our method retains over 95% of its model utility, whereas some baseline methods completely lose their utility. We summarize our key contributions as follows:

- We consider a practical yet challenging LLM continual unlearning setting, conducting theoretical and experimental studies on the unique issues of *accumulative decline* and *cascading degradation* in this context.
- We introduce a novel method, ALKN, which adaptively modulates unlearning gradients to effectively minimize the utility degradation that is particularly severe in LLM continual unlearning scenarios.
- To rigorously evaluate unlearning methods, we construct an evaluation corpus, TRAVIS, and perform extensive experiments across multiple benchmark datasets. The experimental results validate the effectiveness and demonstrate the superiority of ALKN compared to state-of-the-art baselines.

2. Problem Setup and Preliminary

In this section, we formulate the setting of continual unlearning, introduce the baseline methods GA and task vectors. We also demonstrate the intrinsic causes and negative impact of the cascading degradation issue.

2.1. LLM continual unlearning

LLM unlearning. This paper explores the concept of unlearning in causal LLMs. For a model π_θ, where θ denotes the model parameters, the probability of the $(l+1)$-th token given the first l tokens of an input text x is computed by the model followed by a softmax function, formalized as $\pi_\theta(x_{[l+1]}|x_{[:l]})$. The objective of LLM unlearning is to eliminate the influence of a specified target dataset,

$D_u = \{x_u^i\}_{i\in[n_u]}$, along with any related generative capabilities, from the model (Blanco-Justicia et al., 2024). Sample x_u may take the form of plain text or question-answer pairs. Crucially, the unlearning process needs to preserve the knowledge and utility of the model on other data.

LLM continual unlearning. In the context of LLM continual unlearning, the goal extends to managing a sequence of unlearning tasks. Beginning with an initial LLM characterized by pre-trained parameters $\theta^0 \in \mathbb{R}^d$, the model is required to sequentially unlearn datasets $\{D_u^t\}_{t=1}^T$, where each $D_u^t = \{x_u^{t,i}\}_{i\in[n_u^t]}$ corresponds to the t-th unlearning task. During the training of a specific task, the data of other tasks is not accessible. LLM continual unlearning faces the issue of severe degradation of the model utility on non-target data during successive unlearning processes. To safeguard the general utility of the model throughout this iterative process, a retain set $D_r = \{x_r^i\}_{i\in[n_r]}$ can be employed to mitigate unintended degradation of utility.

2.2. Gradient Ascent and Task Vectors

A straightforward approach to addressing the continual unlearning of LLMs is to directly apply existing LLM unlearning methods for each task, such as leveraging gradient ascent-based techniques or task vector methods.

Gradient Ascent. GA is one of the most fundamental fine-tuning-based approaches for unlearning and often serves as a baseline or a foundational module in many existing methods (Yao et al., 2023). While the model acquires knowledge during training through gradient descent, GA facilitates the unlearning process by optimizing in the opposite direction. This is achieved by minimizing the negative cross-entropy loss of predicting the next token on the unlearning dataset:

$$\min_{\theta} -\frac{1}{n_u} \sum_{x\in D_u} \sum_{l} -\log \pi_\theta(x_{[l+1]}|x_{[:l]}). \tag{1}$$

For simplicity, the normalization of the loss by token length is omitted. Unlike gradient descent loss, loss of GA lacks a lower bound, which hinders its ability to converge. Training with GA could lead to significant deviations in model parameters from their pre-training values, which severely degrades model utility. The NPO method is also a gradient ascent-based approach. Compared to GA, the gradient in NPO is scaled by a regularization term that gradually diminishes over the course of training, alleviating the problem of over-unlearning to some extent (Zhang et al., 2024a).

Task Vectors. Introduced by Ilharco et al. (2023), the task vector method manipulates the ability of a pre-trained model to address downstream tasks through fine-tuning and the arithmetic operation of model parameters. When applied to LLM unlearning, the method enables the selective removal of knowledge related to a specific dataset D_u. Specifically, a pre-trained model π_θ is fine-tuned on the unlearning dataset D_u, resulting in a model $\pi_{\theta_{\text{ft}}}$ with enhanced knowledge of D_u. The difference between the fine-tuned parameters and the original parameters, $\theta_{\text{ft}} - \theta$, constitutes the task vector for D_u. By subtracting this task vector from the original parameters θ, a modified model is obtained that has unlearned the knowledge from D_u. Additionally, a scalar λ is introduced to control the magnitude of the task vector:

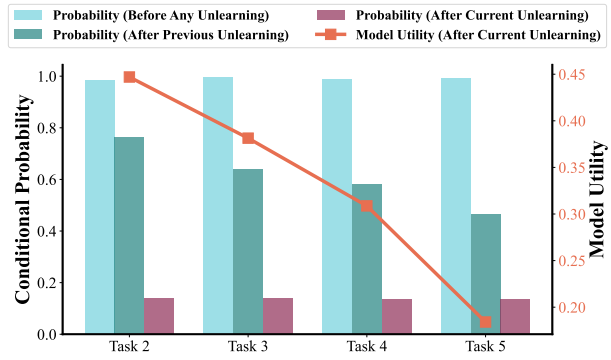
$$\theta_{\text{unlearn}} = \theta - \lambda(\theta_{\text{ft}} - \theta). \tag{2}$$

We theoretically analyze the convergence of the GA and task vector method in Appendix B. In the case of logistic regression unlearning, GA maintains a gradient magnitude above a positive constant and fails to converge. In contrast, the task vector method is proven to converge, indicating that it is less prone to over-unlearning and may cause less utility degradation in LLMs. While the task vector method mitigates utility loss to some extent during a single unlearning task, it remains susceptible to accumulative decline and cascading degradation issues in continual unlearning scenarios.

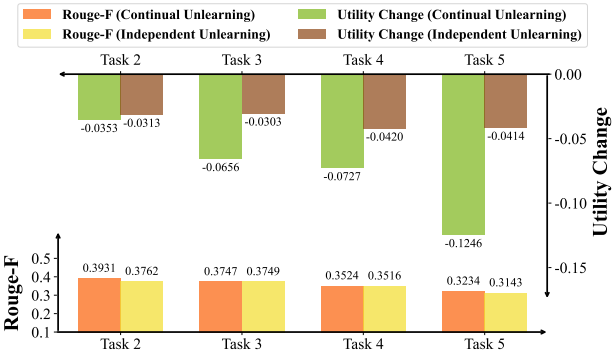
2.3. Cascading Degradation

LLM continual unlearning encounters more complex challenges than one-time unlearning, one of which is cascading degradation, where the interaction between unlearning tasks amplifies the utility degradation. To elaborate, the utility decline during preceding unlearning tasks can also impair the memory retention of the LLM regarding the data of subsequent tasks. Besides, in real-world applications, unlearning data is often highly homogeneous, with requests involving similar content.As seen in Figure 1, data from different users may share similar structures and content. This homogeneity intensifies the impact of preceding unlearning tasks on subsequent tasks. Consequently, the model partially forgets data of subsequent tasks before explicit unlearning, yet these tasks still need to be performed. Unlearning such partially forgotten data with the same optimization process as preceding tasks can easily lead to over-unlearning, amplifying the utility degradation. We refer to this phenomenon as cascading degradation.

To empirically validate the causes and harm of the cascading degradation, we conducted experiments using a baseline approach, NPO+RT on the TOFU dataset, where the LLM model was made to sequentially unlearn five tasks. As shown in Figure 2(a), we examined the model's prediction probabilities $P(y|x)$ on the data of the t-th task before unlearning it (■ Probability After Previous Unlearning). The results reveal a substantial decrease in these probabilities compared to their initial values (■ Probability Before Any Unlearning), indicating that the model had already partially forgotten the t-th task after unlearning the first $t-1$ tasks. To validate the detrimental effects of cascading degradation, we compared continual unlearning with independent



(a) Prediction probability under three conditions and model utility.



(b) Rouge-F and utility change under two unlearning scenarios.

Figure 2. The comparative experiments on the TOFU dataset using the baseline method to sequentially unlearn five disjoint subdatasets. (a) The unlearning of preceding tasks results in the partial forgetting of subsequent tasks, as reflected in the reduced conditional probabilities of task data predicted by the model. (b) Under comparable unlearning performance, continual unlearning exhibits significantly greater utility degradation than independent unlearning due to the effects of cascading degradation.

unlearning, where the model independently unlearns the data of each task starting from pre-trained parameters, as shown in Figure 2(b). The utility changes in the figure denote utility declines in one unlearning task. We ensure that the level of unlearning (Rouge-F) achieved by independent unlearning is similar to that of continual unlearning, enabling a fair comparison of the changes in model utility. Under equivalent conditions, continual unlearning with cascading degradation leads to a greater and increasingly severe decline in model utility compared to independently unlearning each task. We also compare the performance of joint unlearning with continual unlearning in Appendix H to further validate the harm of cascading degradation.

To theoretically substantiate the above conclusion, we consider a toy example where a logistic model $\pi(y=1|x,\theta) = \sigma(\langle x,\theta\rangle)$ performs unlearning with the GA algorithm on a binary classification problem with inputs being x and labels being y. Suppose there are two unlearning tasks with their dataset being correlated. Our objective is to verify whether the first unlearning task influences the second task.

Proposition 2.1. *Consider two correlated datasets D^f and D^s, each composed of $\{(x_i, y_i)\}_{i=1}^n$, satisfying: $\max_{i\neq j} |\langle x_j^f, x_i^s\rangle| < k\langle x_i^f, x_i^s\rangle$, and $y_i^f = y_i^s$, where k is a positive constant. The logistic model starts with an initial parameter θ^0. We compare two optimization scenarios:*

1. *The model successively unlearns on D^s and D^f, yielding intermediate and final parameters: θ_s and θ_{CUL}.*
2. *Starting from θ^0, the model directly unlearns on D^s, yielding parameters θ_{IUL}.*

The parameter changes in the two scenarios during unlearning on D^s satisfy:

$$|| \triangle \theta_{CUL}||_{X^\top X} > || \triangle \theta_{IUL}||_{X^\top X} + TC\sqrt{n}, \quad (3)$$

where C is a positive constant depending on the datasets, T is iteration steps, and n is the number of samples.

Please refer to Appendix C for the proof. Proposition 2.1 demonstrates that during the process of continual unlearning, conducting a preceding unlearning task causes subsequent related tasks to experience larger parameter changes in their own unlearning. In the context of LLMs, this phenomenon may exacerbate the decline in model utility.

3. Method

In continual unlearning for LLMs, there is severe deterioration in model utility. To effectively mitigate the utility decline, the key is reducing the overall changes to the model's parameters under the condition of effective unlearning. Utilizing the task vector method can mitigate utility decline to some extent, but the accumulative decline and cascading degradation issues still exist. Therefore, we further propose three modules to alleviate these issues. *Entropic soft label loss* utilizes generated labels to flexibly adjust the unlearning objective. *Dynamic gradient sparsity* applies a learnable mask to the gradients and encourages different tasks to localize distinct model parameters. *Adaptive parameter modulation* employs a lower learning rate for parameters related to already unlearned tasks. These modules collaboratively address the challenges of **accumulative decline** and **cascading degradation** by mitigating task interference and adaptively regulating the intensity of unlearning. The diagram of these modules is shown in Figure 3. The overall procedure of training is in Algorithm 1.

3.1. Fine-tuning with Entropic Soft Labels

To perform the t-th unlearning task, we employ knowledge negation from the current model $\pi_{\theta^{t-1}}$ with the task vector

method. This process begins by initializing a model with parameters θ^{t-1}, which is then fine-tuned on the unlearning dataset D_u^t. The fine-tuning step yields the intermediate model $\pi_{\theta_{\text{ft}}^t}$, reinforcing the knowledge of the model specific to the t-th dataset (Ilharco et al., 2023). Subsequently, the knowledge associated with the t-th task is removed using model weight arithmetic: $\theta^t = \theta^{t-1} - \lambda(\theta_{\text{ft}}^t - \theta^{t-1})$.

The fine-tuning process for task vector training typically leverages gradient descent optimization with a cross-entropy loss, which quantifies the discrepancy between the next-token prediction labels and the model's output probability distribution conditioned on preceding tokens:

$$\mathcal{L}_{CE}(\theta_{\text{ft}}^t, x) = \sum_l -\log \pi_{\theta_{\text{ft}}^t}(x_{[l+1]}|x_{[:l]}), \tag{4}$$

where $x \in D_u^t$. In the remainder of this paper, $\pi_\theta(x_{[l+1]}|x_{[:l]})$ is denoted as π_θ^* for simplicity.

However, employing cross-entropy loss in continual unlearning scenarios poses the risk of over-unlearning. As demonstrated in Section 2.3, interaction between unlearning tasks could result in a task being partially forgotten even before its data is explicitly unlearned. The predicted probability $\pi_{\theta_{\text{ft}}^t}^*$ for the data of the task gets notably low. In such cases, the cross-entropy loss, which enforces alignment between the output of the model and a one-hot label in the token space, produces gradients as follows:

$$\nabla_{\theta_{\text{ft}}^t}\mathcal{L}_{CE} \propto -\frac{\nabla_{\theta_{\text{ft}}^t}\pi_{\theta_{\text{ft}}^t}^*}{\pi_{\theta_{\text{ft}}^t}^*}. \tag{5}$$

When $\pi_{\theta_{\text{ft}}^t}^*$ is low, the gradient magnitude may increase, causing large parameter updates that negatively impact the utility of the LLM. This issue is theoretically illustrated in Proposition 2.1. Besides, one-hot labels in cross-entropy loss are unnecessarily difficult objectives to optimize towards since the goal of task vector fine-tuning is only to enhance the knowledge of the model regarding the target data. Training with one-hot labels forces redundant parameter adjustments of the LLM model, leading to significant utility decline.

To mitigate this issue, we propose Entropic Soft Labels (ESL) to preserve model utility during task vector fine-tuning. We introduce soft labels that increase the entropy of the target distribution in the loss. The soft label $\tilde{y}$ is derived from the initial predicted probability $\pi_{\theta^{t-1}}^*$ for a training sample and is defined as $\tilde{y} = \sigma\left(s(\pi_{\theta^{t-1}}^* - 1)\right)$, where σ is the sigmoid function and s is a scaling factor that controls the entropy of labels. By leveraging these soft labels, the cross-entropy loss is reformulated as:

$$\mathcal{L}_{ESL}(\theta_{\text{ft}}^t, x) = \sum_l -\tilde{y}_{[l]} \log \pi_{\theta_{\text{ft}}^t}(x_{[l+1]}|x_{[:l]}), \tag{6}$$

where $\tilde{y}_{[l]}$ represents the soft label for the next token. $\mathcal{L}_{ESL}$ mitigates the cascading degradation issue by generating adaptive optimization goals for partially forgotten data.

3.2. Dynamic Gradient Sparsity

In continual unlearning scenarios, we aim to fine-tune different sets of model parameters for each task. This approach helps prevent cumulative parameter divergence from their initial values as the number of tasks increases, thereby reducing the risk of accumulative decline. Moreover, minimizing changes to model parameters during unlearning is essential for maintaining utility. We strive to identify parameters that significantly influence the unlearning objective and focus on adjusting these parameters during training. This would enable effective unlearning while making as few adjustments to the model parameters as possible.

To achieve these goals, we propose to apply learnable masks to model gradients during the fine-tuning of the model θ_{ft}^t on the unlearning set D_u^t, allowing for precise and efficient parameter adjustments. Inspired by von Oswald et al. (2021), for the t-th task, the optimization of the model parameters is given by:

$$\theta_{\text{ft}}^{t,k+1} = \theta_{\text{ft}}^{t,k} - \hat{\alpha}^t \odot (\mathbb{1}_{m^t \geq \eta} \odot \nabla_{\theta_{\text{ft}}}\mathcal{L}_{ESL}(\theta_{\text{ft}}^{t,k}, D_u^t)), \tag{7}$$

where $m^t \in \mathbb{R}^d$ represents a dynamic underlying vector, from which a binary mask is derived using the threshold function $\mathbb{1}_{m^t \geq \eta}$. This function outputs 1 for parameters where the corresponding value of m^t exceeds the threshold η; otherwise it outputs 0. The threshold η is determined by the percentile of m^t and increases gradually throughout the training process. Additionally, $\hat{\alpha}^t$ denotes the adaptive learning rate, which is detailed in Section 3.3.

Mask Value Updating. The application of gradient masking is designed to achieve two purposes: (I) To encourage different tasks to adjust distinct sets of model parameters during unlearning, thereby reducing the overall parameter changes. (II) To dynamically identify model parameters that are important to the unlearning objective while also being beneficial to utility preservation. To achieve these goals, the underlying vector m^t is dynamically learned from the data, allowing the mask to adapt to task-specific requirements during training. For objective (I), the current mask is encouraged to differ from the masks used in previous tasks. Specifically, let $M^{t-1} = \bigcup_{i=1}^{t-1} \mathbb{1}_{m^i \geq \eta}$ represent the set of parameters that underwent substantial adjustments in the first $t-1$ tasks. The mask for the current task is encouraged to minimize its overlap with M^{t-1}. For objective (II), the mask is refined to preserve model utility. The cross-entropy loss on D_r is used as a proxy to evaluate the retained utility of models. The underlying vector m^t of the mask is optimized to minimize this utility loss. The sum of these two objectives with a scaling factor μ is given by:

$$\min_{m^t} \mathcal{L}_{CE}(\theta^{t-1} - \lambda(\theta_{\text{ft}}^{t,k+1} - \theta^0), D_r) + \mu \mathbb{1}_{m^t \geq \eta} \cdot M. \tag{8}$$

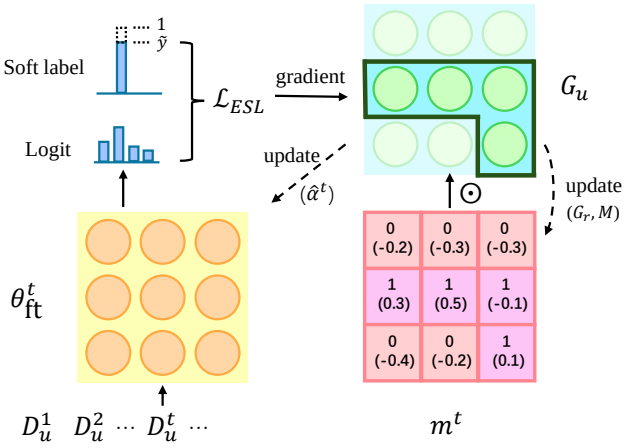


Figure 3. The diagram illustrating the training process of the intermediate model $\pi_{\theta_{\text{ft}}^t}$ for the t-th unlearning task. The training loss $\mathcal{L}_{ESL}$ is computed using entropic soft labels (Section 3.1). Gradients G_u derived from the loss are masked using a binary mask with an underlying vector m^t, dynamically updated based on G_u, G_r and M (Section 3.2). The masked gradients are then employed to update the model parameters θ_{ft}^t with an adaptive learning rate $\hat{\alpha}^t$ (Section 3.3).

By substituting into Equation 7 and differentiating this objective with respect to m^t, we obtain the iterative update rule for m^t:

$$\begin{aligned} m^t \leftarrow & m^t + \lambda G_u \odot G_r - \mu M, \\ \text{where } & G_u = \nabla_{\theta_{\text{ft}}} \mathcal{L}_{ESL}(\theta_{\text{ft}}^{t,k}, D_u^t), \\ & G_r = \nabla_{\theta^{t-1}} \mathcal{L}_{CE}(\theta^{t-1}, D_r). \end{aligned} \tag{9}$$

The derivation is detailed in Appendix D. According to the update rule of m^t, the elements will increase where the gradient signs are consistent across both the unlearning objective and the utility objective. And the corresponding parameters will be selectively activated. In other words, the proposed method selectively activates model parameters, aligning the gradient directions of the unlearning and utility preservation objectives. Furthermore, elements that have a greater impact on both objectives—i.e., those with larger gradient magnitudes—lead to faster changes in the corresponding values of m^t, resulting in quicker activation or freezing of the model parameters.

The proposed method localizes distinct parameters for different unlearning tasks. By avoiding cumulative changes to parameters, we alleviate the accumulative decline in utility of the LLM. And the gradient sparsity helps preserve the overall model utility, further mitigating the issue.

In contrast to methods that rely on static model parameter masks, our proposed approach dynamically adjusts the mask throughout the training process. This dynamic tuning ensures that each model parameter is trained within an appropriate range: parameters with minimal impact on the unlearning objective undergo limited adjustments in the early stages and are subsequently masked, while parameters with significant impact receive more extensive training. This strategy enables balanced training across the model, preventing excessive adjustments to all parameters or overly drastic changes to a small subset. Moreover, our method leverages the underlying vector m^t to aggregate the influence of data on the mask over time, mitigating biases that may arise from extreme values in a single training iteration. By accumulating data-driven insights, the mask adapts more robustly to task-specific requirements, enhancing the stability and effectiveness of the unlearning process.

3.3. Adaptive Parameter Modulation

To further mitigate the decline in model utility, we propose adaptive parameter modulation during fine-tuning θ_{ft}^t. The gradient masks obtained from our training process represent the relationship between the model parameters and the features of each task. Consequently, during the training of the current task, we apply a reduced learning rate to the parameters previously activated, thereby preventing the re-unlearning of features already forgotten and alleviating the cascading degradation of utility. Specifically, let $M^{t-1} = \bigcup_{i=1}^{t-1} \mathbb{1}_{m^i \geq \eta}$ denote the model parameters that have undergone significant adjustments in the first $t-1$ tasks. We apply an adaptive learning rate to model parameters while fine-tuning on the t-th task as follows:

$$\hat{\alpha}^t = \alpha_l M^{t-1} + \alpha_h (1 - M^{t-1}), \tag{10}$$

where α_l denotes a low learning rate for model parameters that are crucial for previous tasks and α_h denotes a relatively high learning rate.

4. Experiments

We conduct extensive experiments on various datasets and models to validate the effectiveness of the proposed method.

4.1. Experimental Settings

Datasets and models. We conduct experiments on three datasets: TOFU, MUSE News, and WHP, each representing scenarios relevant to real-world applications such as privacy preservation and copyright protection. To simulate a continual unlearning setting, we divid the data designated for unlearning into multiple disjoint subsets based on the structure of the datasets. These subsets, either from different parts of the same dataset or a mix of datasets, are treated as sequential unlearning tasks. The experimental details for each dataset are as follows. 1) **TOFU** (Maini et al., 2024) dataset consists of generated fictional author information. We group the information of four authors into one unlearning task, resulting in five or more continual unlearning tasks. 2) **MUSE News** (Shi et al., 2024b)

Algorithm 1 Continual unlearning method ALKN

Input: Sequential unlearning datasets $\{D_u^t\}_{t=1}^T$, retain set D_r, initial model parameters θ^0, hyperparameters s and μ, learning rates α_l and α_h, updating steps T_u for each unlearning task.
Initialize the overall mask $M = \mathbf{0}$.
for $t = 1$ **to** T **do**
  Initialize parameters $\theta_{\text{ft}}^{t,0} = \theta^{t-1}$.
  Initialize the underlying vector $m^t = \mathbf{0}$.
  Calculate $G_r = \nabla_{\theta^{t-1}} \mathcal{L}_{CE}(\theta^{t-1}, D_r)$.
  for $k = 1$ **to** T_u **do**
    Update model parameters $\theta_{\text{ft}}^{t,k}$ with Equation 7.
    Update the underlying vector m^t with Equation 9
    Tune η with percentiles of m^t as in Appendix E.2.
  end for
  $\theta^t = \theta^{t-1} - \lambda(\theta_{\text{ft}}^t - \theta^{t-1})$.
  $M = M \cup \mathbb{1}_{m^i \geq \eta}$.
end for
Output: Final parameters θ^T.

comprises BBC news articles and includes four predefined unlearning subsets that serve as continual unlearning tasks. 3) **WHP** (Who's Harry Potter) (Eldan & Russinovich, 2023; Shi et al., 2024b) involves the unlearning of original text from the Harry Potter series. We treat the original text of three Harry Potter books as three separate unlearning tasks. Following recent works (Maini et al., 2024; Jia et al., 2024a), we use Llama-2-7B and Phi-1.5 as target models.

Metrics. Unlearning algorithms require evaluation from two perspectives: unlearning performance and model utility, both measured using various metrics. Metrics for unlearning performance include: 1) **Rouge-L for unlearning (F-Rouge)** measures similarity between the text generated by the model and the correct text (Maini et al., 2024; Shi et al., 2024b). 2) **Probability (F-Prob)** evaluates the conditional probability $P(\text{answer}|\text{question})$ of model outputs. 3) **Membership inference attack (MIA)** calculates Min-K% Prob (Shi et al., 2024a) for unlearning samples to detect whether the post-unlearning model retains the unlearning set. 4) **Forget Quality (FQ)** quantifies the difference in prediction distributions on the forget set between the post-unlearning model and a retrained model (Maini et al., 2024). Metrics for model utility include: 1) **R-Rouge** and 2) **R-Prob** are also applied to the retain set or holdout set to evaluate the model utility. 3) **T-Rouge** calculates the Rouge-L recall of completing sentences in the proposed TRAVIS dataset. Testing on inferred training data provides a more accurate assessment of the impact of unlearning algorithms on the pre-existing knowledge of the model. Additionally, TRAVIS is generated without thematic constraints, resulting in a broad coverage of content, which enables a more comprehensive evaluation of performance. 4) **Model Utility (MU)** represents the harmonic mean of the above metrics calculated across multiple retain sets in the TOFU dataset (Maini et al., 2024), including the T-Rouge. It provides an aggregate measure of the model utility.

Baselines. Given the limited research on LLM continual unlearning, the proposed method is primarily compared with prominent existing unlearning approaches. These algorithms are applied sequentially to unlearn the data for each task, thereby achieving continual unlearning. **Unlearning methods** include GA (Yao et al., 2023), NPO (Zhang et al., 2024a), and Task Vectors (TV) (Ilharco et al., 2023), as introduced earlier. We also include DPO (Zhang et al., 2024a), SO-PO, and SO-NPO (Jia et al., 2024b) methods. Besides, EUL (Chen & Yang, 2023) incorporates lightweight unlearning layers into the model and explicitly considers scenarios involving multiple unlearning tasks. WAGLE (Jia et al., 2024a) optimizes only the parameters that are crucial to unlearning and retaining objectives during the unlearning process. When applying these methods in continual unlearning scenario, the learning rate is tuned progressively to mitigate catastrophic collapse as the number of tasks increases. **Retaining objectives** are also leveraged with an auxiliary retain dataset to help preserve the model utility in addition to conducting unlearning optimization on the target data. For example, the Retain Loss (RT) is employed as $-\mathbb{E}_{D_r} \log \pi_\theta(x_{[l+1]}|x_{[:l]})$, where gradient descent is performed on the retain dataset to optimize the model. Similarly, KL divergence loss (KL), defined as $\mathbb{E}_{D_r} D(\pi_\theta(\cdot|x)||\pi_{\text{ref}}(\cdot|x))$, is used to ensure that the logits of the model outputs on the retain dataset remain similar to those of the initial model π_{ref} (Zhang et al., 2024a).

4.2. Experiments on Privacy-Preservation Unlearning

In the TOFU dataset, unlearning algorithms are tested with personal information of fictitious authors, where each unlearning task corresponds to different authors. Table 1 presents the unlearning and utility performance of various methods across five continual tasks, showing that utility declines to varying degrees as tasks increase. A Rouge score of approximately 0.35 on the unlearning set is considered sufficient for effective unlearning (Maini et al., 2024). NPO+RT achieves reasonable unlearning performance in the first four tasks but suffers over-unlearning in the last, with model utility dropping below 0.2. Conversely, GA+RT struggles with insufficient unlearning in earlier tasks, followed by over-unlearning later. NPO, EUL, and WAGLE exhibit severe under-unlearning, with F-Rouge reductions of less than 15% in the first task compared to the original model. Although NPO and EUL achieve the highest utility in the third and fourth tasks, this stems from insufficient unlearning. GA+KL, NPO+KL, and SKU avoid over-unlearning but suffer progressively lower utility. In contrast, our method effectively balances unlearning and utility, achieving suf-

Table 1. Unlearning performance (F-Rouge, FQ) and model utility (MU) on the TOFU dataset for the Phi model.

Methods	Task 1			Task 2			Task 3			Task 4			Task 5		
	F-Rouge↓	FQ↑	MU↑	F-Rouge↓	FQ↑	MU↑	F-Rouge↓	FQ↑	MU↑	F-Rouge↓	FQ↑	MU↑	F-Rouge↓	FQ↑	MU↑
Original	0.9614	3.2e-16	0.5237	0.9723	7.4e-16	0.5237	0.9827	4.2e-15	0.5237	0.9895	7.5e-16	0.5237	0.9847	5.0e-17	0.5237
GA+RT	0.5723	7.9e-14	0.4766	0.4047	9.5e-13	0.4427	**0.3524**	1.3e-4	0.3494	**0.3095**	6.5e-5	0.2253	**0.1027**	**9.7e-2**	0.0812
GA+KL	0.3963	7.8e-4	0.3955	0.3896	1.5e-5	0.4270	0.3923	9.1e-3	0.3578	0.4311	5.2e-5	0.2898	0.3108	5.9e-4	0.2059
NPO	0.8430	1.9e-16	0.4836	0.7322	4.6e-15	0.4675	0.6091	8.5e-9	**0.4651**	0.4893	5.4e-9	0.4257	0.4033	8.9e-6	0.3817
NPO+RT	0.3803	9.4e-3	0.4823	0.3931	7.6e-5	0.4470	0.3747	4.8e-4	0.3814	0.3524	6.8e-3	0.3087	0.3234	7.9e-3	0.1841
NPO+KL	0.4109	6.0e-5	0.4729	0.4359	1.0e-6	0.4465	0.4018	6.3e-4	0.3957	0.3891	9.9e-6	0.3350	0.3651	8.5e-5	0.2789
DPO+RT	0.4510	8.9e-10	0.5043	0.4091	4.3e-8	0.4432	0.3983	4.6e-5	0.3602	0.3821	3.1e-4	0.3406	0.3771	3.9e-3	0.3359
TV	0.3957	4.8e-4	0.4825	0.3719	5.2e-3	0.4578	0.3573	8.9e-3	0.3866	0.3371	**9.8e-3**	0.3253	0.3028	4.2e-2	0.2963
SKU	0.4279	3.0e-5	0.4927	0.4055	5.8e-4	0.4100	0.3839	6.0e-4	0.3330	0.3805	6.5e-5	0.3024	0.3346	7.2e-3	0.2216
EUL	0.7885	2.4e-15	0.4839	0.5562	2.9e-12	0.4662	0.5264	7.2e-12	0.4551	0.5171	1.6e-13	0.4537	0.5058	8.0e-12	0.3938
SO-NPO	0.4525	7.3e-8	0.4935	0.4316	6.2e-6	0.4532	0.4168	9.9e-7	0.3725	0.4153	5.0e-5	0.3472	0.4002	4.1e-4	0.3001
WAGLE	0.7667	2.0e-15	0.4731	0.5031	4.8e-9	0.4669	0.4593	7.7e-6	0.4314	0.3860	2.5e-5	0.3400	0.3731	3.5e-4	0.3013
Ours w/o ESL	0.3627	3.5e-2	0.5016	0.3320	3.7e-2	0.4413	0.3067	7.3e-2	0.4170	0.2684	1.4e-1	0.3807	0.2344	5.7e-1	0.3351
Ours w/o DGS	0.3827	5.4e-3	0.5048	0.3642	1.6e-3	0.4559	0.3536	6.2e-3	0.4437	0.3297	4.0e-3	0.4296	0.3073	7.9e-1	0.3893
Ours w/o APM	0.3703	1.4e-2	0.5128	0.3464	8.2e-3	0.4784	0.3320	4.7e-2	0.4524	0.3043	7.3e-2	0.4367	0.2517	2.5e-1	0.3768
Ours	**0.3703**	**1.4e-2**	**0.5128**	**0.3671**	**2.0e-3**	**0.4853**	0.3681	**2.1e-3**	0.4627	0.3452	1.3e-4	**0.4578**	0.3314	1.1e-2	**0.4534**

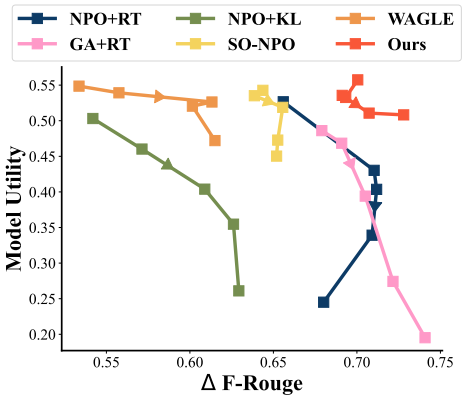


Figure 4. Model utility (MU) and changes of F-Rouge of five unlearning tasks on TOFU dataset for the Llama2 model.

ficient unlearning while minimizing the decline in model utility. Notably, it not only achieves the best performance across all metrics on the first two tasks but also surpasses baseline methods from the perspective of multiple tasks. After completing all five unlearning tasks, our method exhibits less than a 0.06 reduction in utility, effectively mitigating the cumulative utility degradation typically observed in continual unlearning scenario. This highlights its significantly lower impact on utility compared to other approaches.

Ablation study. We present the ablation study results in Table 1, illustrating the impact of each module on the overall performance of the model. When the entropic soft label (ESL) module is removed, the model exhibits significant cascading degradation, resulting in severe utility decline during the unlearning of later tasks. The dynamic gradient sparsity (DGS) module plays a critical role in identifying model parameters that balance utility retention and effective unlearning. Without the DGS module, the overall model utility decreases, and the cumulative utility degradation becomes more pronounced. Similarly, the absence of the adaptive parameter modulation (APM) module leads to a sustained decline in utility over the course of unlearning.

Figure 4 illustrates the unlearning results for the Llama2 model on the TOFU dataset. The persistent issue of utility degradation is evident, as the data points of the same color representing the performance of successive tasks are generally arranged sequentially from up to down, reflecting the cumulative effect of unlearning tasks on model utility. For the latter tasks, NPO+RT and GA+RT suffer a drastic decline in model utility. While WAGLE exhibits poor unlearning effectiveness with low changes of F-Rouge. Notably, the data points corresponding to our method are clustered more tightly across the five tasks, indicating a more stable and controlled utility decline in the continual unlearning scenario. Furthermore, these points are positioned nearer to the upright corner of the plot compared to other methods, signifying a superior trade-off between effective unlearning and maintaining model utility. By achieving a high degree of unlearning with minimal impact on utility, our approach demonstrates clear advantages over baseline methods.

4.3. Experiments on Copyright-Protection Unlearning

The WHP and MUSE News datasets are designed to evaluate the unlearning of verbatim memory of news articles and Harry Potter books by LLMs. We conducted experiments with the Llama2 model independently on each dataset and further explored the effects of alternating tasks between the two datasets. Please refer to Appendix H for more results.

The results on the WHP dataset are shown in Table 2 and Figure 7 (b). Utilizing the `TRAVIS` dataset provides the evaluation of general utility wider than domain knowledge in retain dataset. In the WHP dataset, the sequential tasks involve unlearning data from several books in the Harry Potter series. Due to the strong interconnections between the data across these tasks, unlearning earlier tasks inevitably impacts subsequent tasks, leading to pronounced cascading degradation. For baseline methods, model utility declines largely after completing the unlearning. In contrast, our method effectively mitigates the cascading degradation is-

Table 2. Final model utility metrics and average unlearning performance metrics over three tasks on the WHP dataset.

Methods	Unlearning		Utility	
	F-Rouge↓	MIA↓	R-Rouge↑	T-Rouge↑
Original	0.9821	0.5206	0.6940	0.7643
NPO+RT	**0.1559**	0.2607	0.0	0.0
NPO+KL	0.1620	**0.2452**	0.0	0.0
GA+RT	0.1855	0.2674	0.1172	0.1975
SO-NPO	0.3080	0.3892	0.4109	0.4536
WAGLE	0.2946	0.3745	0.3721	0.4594
Ours	0.1790	0.2729	**0.4586**	**0.5617**

Table 3. Rouge Score Sensitivity to Gaussian Noise in Model Parameters.

Noise Std(%)	TRAVIS	TOFU	WHP
Original	0.7643	0.8965	0.6940
0.1%	0.7643	0.8965	0.6940
0.5%	0.7576	0.8965	0.6940
1.0%	0.7425	0.8965	0.6940
2.0%	0.7320	0.8958	0.6940

sue, demonstrating improved preservation of model utility in continual unlearning scenarios.

4.4. Evaluation of TRAVIS Dataset Sensitivity

The TRAVIS dataset consists of a wide variety of topics, enabling a comprehensive evaluation of the model's overall utility. As the dataset is generated by the LLM, it provides an accurate baseline for evaluating the model's original utility. To investigate the sensitivity of the TRAVIS dataset to variations in model performance, we conducted a synthetic experiment by introducing a small amount of random Gaussian noise to the model parameters to simulate destruction of model utility. We compared the Rouge scores of the TRAVIS dataset against those of the TOFU and WHP datasets to assess their relative sensitivity to model utility changes. As presented in Table 3, the TRAVIS dataset is the most sensitive to changes in model utility. Notably, the TRAVIS Rouge score exhibits a decline when the noise standard deviation reaches 0.5%, whereas the TOFU and WHP datasets remain largely unaffected until higher noise levels.

5. Conclusion

In this study, we explore the challenges of continual unlearning in LLMs, which is a practical and complicated setting. Specifically, the issues of accumulative decline and cascading degradation exacerbate the utility deterioration problem for LLM unlearning methods. To mitigate these issues, we propose `ALKN`, a method that minimizes adjustments to model parameters by identifying the model parameters that are crucial to the knowledge representation of each task and tuning these parameters with adaptive intensity. The proposed method is built on the task vector framework with three novel modules applied in the fine-tuning process, dynamically localizing model parameters and adaptively refining gradients. Besides, we construct an evaluation corpus, `TRAVIS`, designed to test the overall utility of LLMs comprehensively. The experimental results demonstrate that the proposed method effectively preserves the utility of LLMs while accomplishing sufficient unlearning, outperforming various baseline methods.

Acknowledgments

This work is funded by the National Science and Technology Major Project (No. 2022ZD0114903) and the Natural Science Fundation of China (NSFC. No. 62176132). Sen Cui would like to acknowledge the financial support received from Shuimu Tsinghua scholar program. Qizhou Wang and Bo Han were supported by the NSFC General Program No. 62376235, GDST Basic Research Fund Nos. 2022A1515011652 and 2024A1515012399. This research was partially supported by Basic Research Grant (Super AI) of Institute for AI and Beyond of the University of Tokyo.

Impact Statement

We propose a novel method for continual unlearning in large language models that is designated to protect personal privacy and copyright. It can also be utilized to prevent harmful responses from LLMs. For similar reasons, the proposed method also does not involve storing training data. The constructed corpus `TRAVIS` is generated by the Phi model and the Llama2 model, only presenting general knowledge. It is manually filtered to guarantee no private or sensitive data is included.

References

Blanco-Justicia, A., Jebreel, N., Manzanares-Salor, B., Sánchez, D., Domingo-Ferrer, J., Collell, G., and Tan, K. E. Digital forgetting in large language models: A survey of unlearning methods. *CoRR*, abs/2404.02062, 2024.

Brophy, J. and Lowd, D. Machine unlearning for random forests. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1092–1104. PMLR, 2021.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu,

J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Cao, Y. and Yang, J. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pp. 463–480. IEEE Computer Society, 2015.

Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T. B., Song, D., Erlingsson, Ú., Oprea, A., and Raffel, C. Extracting training data from large language models. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pp. 2633–2650. USENIX Association, 2021.

Chen, J. and Yang, D. Unlearn what you want to forget: Efficient unlearning for llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 12041–12052. Association for Computational Linguistics, 2023.

Eldan, R. and Russinovich, M. Who's harry potter? approximate unlearning in llms. *CoRR*, abs/2310.02238, 2023.

Golatkar, A., Achille, A., and Soatto, S. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIX*, volume 12374 of *Lecture Notes in Computer Science*, pp. 383–398. Springer, 2020.

Golatkar, A., Achille, A., Ravichandran, A., Polito, M., and Soatto, S. Mixed-privacy forgetting in deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 792–801. Computer Vision Foundation / IEEE, 2021.

Graves, L., Nagisetty, V., and Ganesh, V. Amnesiac machine learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 11516–11524. AAAI Press, 2021.

Guo, C., Goldstein, T., Hannun, A., and Van Der Maaten, L. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.

Han, B., Niu, G., Yu, X., Yao, Q., Xu, M., Tsang, I. W., and Sugiyama, M. SIGUA: forgetting may make learning with noisy labels more robust. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4006–4016. PMLR, 2020.

Hong, Y., Yu, L., Ravfogel, S., Yang, H., and Geva, M. Intrinsic evaluation of unlearning using parametric knowledge traces. *CoRR*, abs/2406.11614, 2024a.

Hong, Y., Zou, Y., Hu, L., Zeng, Z., Wang, D., and Yang, H. Dissecting fine-tuning unlearning in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 3933–3941. Association for Computational Linguistics, 2024b.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Izzo, Z., Smart, M. A., Chaudhuri, K., and Zou, J. Approximate data deletion from machine learning models. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2008–2016. PMLR, 2021.

Ji, J., Liu, Y., Zhang, Y., Liu, G., Kompella, R. R., Liu, S., and Chang, S. Reversing the forget-retain objectives: An efficient LLM unlearning framework from logit difference. *CoRR*, abs/2406.08607, 2024.

Jia, J., Liu, J., Zhang, Y., Ram, P., Baracaldo, N., and Liu, S. WAGLE: strategic weight attribution for effective and modular unlearning in large language models. *CoRR*, abs/2410.17509, 2024a.

Jia, J., Zhang, Y., Zhang, Y., Liu, J., Runwal, B., Diffenderfer, J., Kailkhura, B., and Liu, S. SOUL: unlocking the power of second-order optimization for LLM unlearning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 4276–4292. Association for Computational Linguistics, 2024b.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Liu, S., Yao, Y., Jia, J., Casper, S., Baracaldo, N., Hase, P., Xu, X., Yao, Y., Li, H., Varshney, K. R., Bansal, M., Koyejo, S., and Liu, Y. Rethinking machine unlearning for large language models. *CoRR*, abs/2402.08787, 2024a.

Liu, Z., Dou, G., Tan, Z., Tian, Y., and Jiang, M. Towards safer large language models through machine unlearning. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 1817–1829. Association for Computational Linguistics, 2024b.

Lynch, A., Guo, P., Ewart, A., Casper, S., and Hadfield-Menell, D. Eight methods to evaluate robust unlearning in llms. *CoRR*, abs/2402.16835, 2024.

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., and Kolter, J. Z. TOFU: A task of fictitious unlearning for llms. *CoRR*, abs/2401.06121, 2024.

Niu, G., du Plessis, M. C., Sakai, T., Ma, Y., and Sugiyama, M. Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 1199–1207, 2016.

Rozière, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., Adi, Y., Liu, J., Remez, T., Rapin, J., Kozhevnikov, A., Evtimov, I., Bitton, J., Bhatt, M., Canton-Ferrer, C., Grattafiori, A., Xiong, W., Défossez, A., Copet, J., Azhar, F., Touvron, H., Martin, L., Usunier, N., Scialom, T., and Synnaeve, G. Code llama: Open foundation models for code. *CoRR*, abs/2308.12950, 2023.

Schmidt, M. Cpsc 540: Machine learning. *Convergence of Gradient Descent*, 2017.

Scholten, Y., Günnemann, S., and Schwinn, L. A probabilistic perspective on unlearning and alignment for large language models. *CoRR*, abs/2410.03523, 2024.

Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., and Zettlemoyer, L. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a.

Shi, W., Lee, J., Huang, Y., Malladi, S., Zhao, J., Holtzman, A., Liu, D., Zettlemoyer, L., Smith, N. A., and Zhang, C. MUSE: machine unlearning six-way evaluation for language models. *CoRR*, abs/2407.06460, 2024b.

Thaker, P., Hu, S., Kale, N., Maurya, Y., Wu, Z. S., and Smith, V. Position: LLM unlearning benchmarks are weak measures of progress. *CoRR*, abs/2410.02879, 2024.

Ullah, E., Mai, T., Rao, A., Rossi, R. A., and Arora, R. Machine unlearning via algorithmic stability. In *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*, volume 134 of *Proceedings of Machine Learning Research*, pp. 4126–4142. PMLR, 2021.

von Oswald, J., Zhao, D., Kobayashi, S., Schug, S., Caccia, M., Zucchet, N., and Sacramento, J. Learning where to learn: Gradient sparsity in meta and continual learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 5250–5263, 2021.

Wang, Q., Han, B., Yang, P., Zhu, J., Liu, T., and Sugiyama, M. Unlearning with control: Assessing real-world utility for large language model unlearning. *CoRR*, abs/2406.09179, 2024a.

Wang, Q., Lin, Y., Chen, Y., Schmidt, L., Han, B., and Zhang, T. A sober look at the robustness of clips to spurious features. In *Advances in Neural Information Processing Systems*, 2024b.

Wang, Q., Han, B., Yang, P., Zhu, J., Liu, T., and Sugiyama, M. Towards effective evaluations and comparison for llm unlearning methods. In *International Conference on Learning Representations*, 2025a.

Wang, Q., Zhou, J. P., Zhou, Z., Shin, S., Han, B., and Weinberger, K. Q. Rethinking llm unlearning objectives: A gradient perspective and go beyond. In *International Conference on Learning Representations*, 2025b.

Wang, Y., Wang, Q., Liu, F., Huang, W., Du, Y., Du, X., and Han, B. Gru: Mitigating the trade-off between unlearning and retention for large language models. In *International Conference on Machine Learning*, 2025c.

Wu, R., Yadav, C., Salakhutdinov, R., and Chaudhuri, K. Evaluating deep unlearning in large language models. *CoRR*, abs/2410.15153, 2024.

Wu, Y., Dobriban, E., and Davidson, S. B. Deltagrad: Rapid retraining of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10355–10366. PMLR, 2020.

Yang, P., Wang, Q., Huang, Z., Liu, T., Zhang, C., and Han, B. Exploring criteria of loss reweighting to enhance llm unlearning. In *International Conference on Machine Learning*, 2025.

Yao, Y., Xu, X., and Liu, Y. Large language model unlearning. *CoRR*, abs/2310.10683, 2023.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. In *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018*, pp. 268–282. IEEE Computer Society, 2018.

Yuan, X., Pang, T., Du, C., Chen, K., Zhang, W., and Lin, M. A closer look at machine unlearning for large language models. *CoRR*, abs/2410.08109, 2024.

Zhang, H., Nakamura, T., Isohara, T., and Sakurai, K. A review on machine unlearning. *SN Comput. Sci.*, 4(4): 337, 2023.

Zhang, R., Lin, L., Bai, Y., and Mei, S. Negative preference optimization: From catastrophic collapse to effective unlearning. *CoRR*, abs/2404.05868, 2024a.

Zhang, Z., Zhang, J., Yao, H., Niu, G., and Sugiyama, M. On unsupervised prompt learning for classification with black-box language models. *CoRR*, abs/2410.03124, 2024b.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J., and Wen, J. A survey of large language models. *CoRR*, abs/2303.18223, 2023.

A. Related Work

Machine unlearning studies the problem of efficiently removing specific data from machine learning models, ensuring compliance with privacy regulations like GDPR (Cao & Yang, 2015; Wang et al., 2025c; Zhang et al., 2023; Niu et al., 2016; Zhang et al., 2024b). Machine unlearning methods are generally categorized into exact unlearning (Ullah et al., 2021; Brophy & Lowd, 2021) and approximate unlearning. The latter modifies the model or dataset to simulate the removal effect without full retraining (Wu et al., 2020; Golatkar et al., 2020; 2021; Graves et al., 2021). There is a line of work that employs Newton's method to remove the influence of target data involving calculating second-order derivatives of model parameters (Koh & Liang, 2017; Guo et al., 2019; Izzo et al., 2021). However, the approach is computationally prohibitive for LLMs. In addition, precisely evaluating LLM unlearning algorithms, as is commonly done in the field of machine unlearning, is often impractical because obtaining a fully retrained model for comparison is rarely feasible (Scholten et al., 2024).

Among the studies on LLM unlearning, some focus on establishing evaluation metric (Wang et al., 2024a; 2025b; Yang et al., 2025; Lynch et al., 2024; Yuan et al., 2024; Wu et al., 2024), while others emphasize the design of effective algorithms. With trivial unlearning methods, the process often leads to excessive unlearning, where the performance of the model on non-target data is unintentionally degraded. The primary objective of LLM unlearning is to preserve the model utility on normal data while effectively unlearning the target information. NPO is proposed to mitigate the issue of unbounded optimization loss of GA (Zhang et al., 2024a). Eldan & Russinovich (2023) suggest replacing sensitive keywords with generic terms, enabling the model to produce generic predictions through reinforcement bootstrapping. The constructed generic outputs enable unlearning by fine-tuning the LLM model on them. Ji et al. (2024) propose ULD, which reverses the optimization direction during the training of an assistant model. SKU incorporates an object function that embeds robustness and utility retention into task vector training (Liu et al., 2024b), offering a more efficient approach to unlearning. WAGLE seeks to locate the specific LLM parameters that are most critical for both unlearning and retaining objectives and fine-tuning exclusively on these parameters (Jia et al., 2024a). Several studies have also examined the impact of unlearning techniques on various modules within transformer models (Hong et al., 2024b;a). Despite the substantial body of research on LLM unlearning, the practical challenge of continual unlearning remains underexplored and warrants further investigation.

B. Convergence of GA and the task vector method

To compare the convergence of GA and the task vector method, we consider a standard binary classification problem with training data $\{(x_i, y_i)\}_{i=1}^n$, where $x \in \mathbb{R}^{d_l}$ and $y \in \{0, 1\}$. A logistic model is used to predict the probabilities $\pi(y = 1|x, \theta) = \sigma(\langle x, \theta \rangle)$, where σ represents the sigmoid function (Schmidt, 2017).

B.1. Convergence of the task vector method

Theorem B.1. *Using the task vector algorithm with a learning rate $0 < \eta < \frac{2}{L}$ to optimize the aforementioned logistic regression problem can guarantee convergence. Let θ^t denotes the parameters after the t-th iteration of optimizing. The loss function $f_{TV}(\theta^t)$ strictly decreases:*

$$f_{TV}(\theta^{t+1}) < f_{TV}(\theta^t),\ t \geq 0, \tag{11}$$

and converges to its optimal value:

$$\lim_{t\to\infty} f_{TV}(\theta^t) = f_{TV}(\theta^*) \tag{12}$$

Proof. The fine-tuning process of the task vector method utilizes the gradient descent algorithm with the following loss function:

$$f_{\text{TV}}(\theta^t) = -\sum_i y_i \log \sigma(\langle x, \theta^t \rangle) - (1 - y_i) \log \sigma(-\langle x, \theta^t \rangle). \tag{13}$$

The function f_{TV} is L-smooth. Thus the following inequality holds:

$$f_{\text{TV}}(\theta^{t+1}) \leq f_{\text{TV}}(\theta^t) + \nabla f_{\text{TV}}(\theta^t)^\top (\theta^{t+1} - \theta^t) + \frac{L}{2}||\theta^{t+1} - \theta^t||^2. \tag{14}$$

Substituting $\theta^{t+1} - \theta^t = -\eta \nabla f_{\text{TV}}(\theta^t)$ yields:

$$f_{\text{TV}}(\theta^{t+1}) \leq f_{\text{TV}}(\theta^t) - \eta \left(1 - \frac{L\eta}{2}\right) ||\nabla f_{\text{TV}}(\theta^t)||^2. \tag{15}$$

With $0 < \eta < \frac{2}{L}$, we get a strict decrease in $f_{\text{TV}}(\theta^t)$:

$$f_{\text{TV}}(\theta^{t+1}) < f_{\text{TV}}(\theta^t) \tag{16}$$

With lower bound $f_{\text{TV}}(\theta) \geq 0$, the loss function converges to its optimal value:

$$\lim_{t\to\infty} f_{\text{TV}}(\theta^t) = f_{\text{TV}}(\theta^*) \tag{17}$$

B.2. Non-convergence of the GA method

Theorem B.2. *Optimizing the aforementioned logistic regression problem using the GA algorithm will result in the norm of the gradient remaining greater than a certain constant:*

$$||\nabla_{\theta^t} f_{GA}(\theta^t)|| \geq \sqrt{C}, \tag{18}$$

where θ^t denotes the parameters after the t-th iteration of optimizing and C is a positive constant only depending on the training data $\{(x_i, y_i)\}_{i=1}^n$ and initial parameters θ^0. And the objective function $f_{GA}(\theta^t)$ strictly decreasing to negative infinity:

$$\begin{aligned} f_{GA}(\theta^t) &= f_{GA}(\theta^0) - t\eta C, \\ \lim_{t\to\infty} f_{GA}(\theta^t) &= -\infty. \end{aligned} \tag{19}$$

Proof. The loss function of GA is:

$$f_{\text{GA}}(\theta^t) = \sum_i y_i \log \sigma(\langle x, \theta^t\rangle) + (1 - y_i) \log \sigma(-\langle x, \theta^t\rangle) \tag{20}$$

The gradient with respect to the parameters θ^t is:

$$\begin{aligned} \nabla_{\theta^t} f_{\text{GA}}(\theta^t) &= \sum_i (2y_i - 1)(1 - \pi_i^t) x_i, \\ \text{where } \pi_i^t &= \sigma(\langle x, \theta\rangle) \text{ when } y_i = 1, \\ \pi_i^t &= \sigma(-\langle x, \theta\rangle) \text{ when } y_i = 0. \end{aligned} \tag{21}$$

We define $m_i^t = (2y_i - 1)(1 - \pi_i^t)$, $\gamma_{i,j} = \langle x_i, x_j\rangle$, and γ_i denotes $(\gamma_{i,1}, \gamma_{i,2}, ..., \gamma_{i,n})$. Besides, we define $q_i^t = \langle m^t, \gamma_i\rangle$.

Inspired by Zhang et al. (2024a), we use the induction method to proceed.

Case 1: $t = 0$. Since $||x_i||_2 \cdot ||\theta^0||_2$ is a finite value, we can derive:

$$\begin{aligned} C_1 \leq m_i^0 \leq 1 \text{ when } y_i = 1, \\ -1 \leq m_i^0 \leq -C_2 \text{ when } y_i = 0, \end{aligned} \tag{22}$$

where C_1 and C_2 are two positive constants. We assume there exists a constant C_0 such that $\max_{i\neq j} |\gamma_{i,j}| \leq \frac{C_0}{n}$, and satisfying:

$$\left| \sum_{i\neq j} m_j^0 \gamma_{i,j} \right| \leq \frac{\gamma_{i,i} |m_i^0|}{2}. \tag{23}$$

We obtain:

$$\begin{aligned} q_i^0 = \langle m^0, \gamma_i\rangle \geq \frac{|m_i|\gamma_{i,i}}{2} > 0 \text{ when } y_i = 1, \\ q_i^0 = \langle m^0, \gamma_i\rangle \leq -\frac{|m_i|\gamma_{i,i}}{2} < 0 \text{ when } y_i = 0. \end{aligned} \tag{24}$$

We set $C_3 = \min(C_1, C_2)$ and $C_4 = \frac{C_3\gamma_{i,i}}{2}$, then we have:

$$\begin{aligned} q_i^0 = \langle m^0, \gamma_i\rangle \geq C_4 > 0 \text{ when } y_i = 1, \\ q_i^0 = \langle m^0, \gamma_i\rangle \leq -C_4 < 0 \text{ when } y_i = 0. \end{aligned} \tag{25}$$

Then we derive m_i^1:

$$m_i^1 = (2y_i - 1)(1 - \pi_i^1) = (2y_i - 1)(1 - \sigma((2y_i - 1)\langle x_i, \theta^1\rangle)). \tag{26}$$

Substracting $\theta^1 = \theta^0 - \eta \nabla_{\theta^t} f_{\text{GA}}(\theta^0) = \theta^0 - \eta \sum_i m_i^0 x_i$, we obtain:

$$m_i^1 = (2y_i - 1)\left(1 - \sigma\left((2y_i - 1)\langle x_i, \theta^0\rangle - \eta(2y_i - 1)q_i^0\right)\right). \tag{27}$$

Comparing it with $m_i^0 = (2y_i - 1)(1 - \sigma((2y_i - 1)\langle x_i, \theta^0\rangle))$, we can derive:

$$\begin{aligned} m_i^1 \geq m_i^0 \text{ when } y_i = 1 \\ m_i^1 \leq m_i^0 \text{ when } y_i = 0. \end{aligned} \tag{28}$$

Case 2: $t = K > 0$. We suppose:

$$\begin{aligned} q_i^t \geq C_4 > 0 \text{ when } y_i = 1, \\ q_i^t \leq -C_4 < 0 \text{ when } y_i = 0, \end{aligned} \tag{29}$$

for $0 \leq t \leq K$. And we suppose:

$$\begin{aligned} m_i^{t+1} \geq m_i^t \geq C_1 \text{ when } y_i = 1 \\ m_i^{t+1} \leq m_i^t \leq -C_2 \text{ when } y_i = 0, \end{aligned} \tag{30}$$

for $0 \leq t \leq K - 1$. With this condition, we can derive the same conclusion for q_i^{K+1} as in Case 1:

$$\begin{aligned} q_i^{K+1} \geq C_4 > 0 \text{ when } y_i = 1, \\ q_i^{K+1} \leq -C_4 < 0 \text{ when } y_i = 0, \end{aligned} \tag{31}$$

and the same conclusion for m_i^{K+1}:

$$\begin{aligned} m_i^{K+1} \geq m_i^K \geq C_1 \text{ when } y_i = 1 \\ m_i^{K+1} \leq m_i^K \leq -C_2 \text{ when } y_i = 0. \end{aligned} \tag{32}$$

From above, we prove that $|q_i^t| \geq C_4$ and $(2y_i - 1)m_i^{t+1} > (2y_i - 1)m_i^t$ for all $t \geq 0$. We can further derive that:

$$f_{\text{GA}}(\theta^{t+1}) \leq f_{\text{GA}}(\theta^t), \tag{33}$$

and

$$||\nabla_{\theta^t} f_{\text{GA}}(\theta^t)||_{X^\top X} = \sqrt{\nabla_{\theta^t} f_{\text{GA}}(\theta^t)^\top X^\top X \nabla_{\theta^t} f_{\text{GA}}(\theta^t)} = \sqrt{(q^t)^\top q^t} \geq C_4\sqrt{n}. \tag{34}$$

By incorporating the condition $||\nabla_{\theta^t} f_{\text{GA}}(\theta^t)||_{X^\top X} \leq \sqrt{\lambda_{\max}}||\nabla_{\theta^t} f_{\text{GA}}(\theta^t)||_2$, we can obtain:

$$||\nabla_{\theta^t} f_{\text{GA}}(\theta^t)||_2 \geq \frac{C_4\sqrt{n}}{\sqrt{\lambda_{\max}}} = \sqrt{C}, \tag{35}$$

where C is a positive constant only depending on the training data $\{(x_i, y_i)\}_{i=1}^n$ and initial parameters θ^0.

Furthermore, since $f_{\text{GA}}(\theta)$ is a concave function, it satisfies the following inequality:

$$f_{\text{GA}}(\theta^{t+1}) \leq f_{\text{GA}}(\theta^t) + \nabla_{\theta^t} f_{\text{GA}}(\theta^t)^\top(\theta^{t+1} - \theta^t) = f_{\text{GA}}(\theta^t) - \eta||\nabla_{\theta^t} f_{\text{GA}}(\theta^t)||_2^2, \tag{36}$$

then $f_{\text{GA}}(\theta^{t+1}) \leq f_{\text{GA}}(\theta^t) - \eta C$, and we can obtain:

$$\lim_{t\to\infty} f_{\text{GA}}(\theta^t) = \lim_{t\to\infty} f_{\text{GA}}(\theta^0) - t\eta C = -\infty \tag{37}$$

C. Proof of Proposition 2.1

As defined for the GA, the loss function and its gradient are given as follows:

$$\begin{aligned} f_{\text{GA}}(\theta) &= \sum_i y_i \log \sigma(\langle x, \theta \rangle) + (1 - y_i) \log \sigma(-\langle x, \theta \rangle) \\ \nabla_\theta f_{\text{GA}}(\theta) &= \sum_i (2y_i - 1)(1 - \pi_i) x_i, \end{aligned} \tag{38}$$

where π_i follows the definition in Appendix B.2, as do q_i, m_i and $\gamma_{i,j}$.

Independently training on D^s.

In this scenario, the model is directly trained on D^s starting from the initial parameter θ^0 for T iterations with a learning rate η, resulting in the updated parameters θ_{IUL}:

$$\begin{aligned} \theta_{\text{IUL}}^{t+1} &= \theta_{\text{IUL}}^t - \eta \sum_i m_{\text{IUL},i}^t x_i^s, \\ \langle \theta_{\text{IUL}}^T, x_i^s \rangle &= \langle \theta^0, x_i^s \rangle - \eta \sum_t q_{\text{IUL},i}^t. \end{aligned} \tag{39}$$

As in Appendix B.2, we can prove that:

$$\begin{aligned} q_{\text{IUL},i}^t &\geq C_4 > 0 \text{ when } y_i = 1, \\ q_{\text{IUL},i}^t &\leq -C_4 < 0 \text{ when } y_i = 0, \end{aligned} \tag{40}$$

Continuously training on D^f and D^s.

In this scenario, the model is first trained on D^f starting from θ^0, yielding parameters θ_f:

$$\theta_{\text{s}}^{t+1} = \theta_{\text{s}}^t - \eta \sum_i m_{\text{s},i}^t x_i^s \tag{41}$$

The training on D^f affects the model predictions on D^s:

$$\langle \theta_f^{t+1}, x_i^s \rangle = \langle \theta_f^t, x_i^s \rangle - \eta \sum_j m_{f,i}^t \langle x_{f,j}, x_{s,i} \rangle \tag{42}$$

Because $\max_{i \neq j} |\langle x_j^f, x_i^s \rangle| < k \langle x_i^f, x_i^s \rangle$, we assume there exists a constant C_0 such that $\max_{i \neq j} |\langle x_j^f, x_i^s \rangle| \leq \frac{C_0}{n}$, and satisfying:

$$\left| \sum_{i \neq j} m_{f,i}^t |\langle x_j^f, x_i^s \rangle| \right| \leq \frac{\langle x_i^f, x_i^s \rangle |m_{f,i}^t|}{2}. \tag{43}$$

Therefore, we can obtain:

$$\begin{aligned} \langle \theta_f^{t+1}, x_i^s \rangle &< \langle \theta_f^t, x_i^s \rangle - \frac{\langle x_i^f, x_i^s \rangle |m_{f,i}^t|}{2} \text{ when } y_i = 1, \\ \langle \theta_f^{t+1}, x_i^s \rangle &> \langle \theta_f^t, x_i^s \rangle + \frac{\langle x_i^f, x_i^s \rangle |m_{f,i}^t|}{2} \text{ when } y_i = 0. \end{aligned} \tag{44}$$

Thus the effect of training on D^f demonstrates as:

$$\begin{aligned} \langle \theta_f, x_i^s \rangle &< \langle \theta^0, x_i^s \rangle - C_1 T \text{ when } y_i = 1, \\ \langle \theta_f, x_i^s \rangle &> \langle \theta^0, x_i^s \rangle + C_1 T \text{ when } y_i = 0. \end{aligned} \tag{45}$$

Then the model starts training on D^s with initial parameters being θ_f, yielding updated parameters θ_{CUL}:

$$\begin{aligned} \theta_{\text{CUL}}^{t+1} &= \theta_{\text{CUL}}^{t} - \eta \sum_i m_{\text{CUL},i}^t x_i^s, \\ \langle \theta_{\text{CUL}}^T, x_i^s \rangle &= \langle \theta_f, x_i^s \rangle - \eta \sum_t q_{\text{CUL},i}^t. \end{aligned} \tag{46}$$

Since $\theta_{\text{CUL}}^0 = \theta_f$ the inequality in Equation 45, we can derive:

$$\begin{aligned} q_{\text{CUL},i}^0 &> q_{\text{IUL},i}^0 + C_2 \text{ when } y_i = 1, \\ q_{\text{CUL},i}^0 &< q_{\text{IUL},i}^0 - C_2 \text{ when } y_i = 0. \end{aligned} \tag{47}$$

Using the induction method as in Appendix B.2, we can prove that:

$$\begin{aligned} q_{\text{CUL},i}^t &> q_{\text{IUL},i}^t + C_2 > 0 \text{ when } y_i = 1, \\ q_{\text{CUL},i}^t &< q_{\text{IUL},i}^t - C_2 < 0 \text{ when } y_i = 0. \end{aligned} \tag{48}$$

Therefore, $|\sum_t q_{\text{CUL},i}^t| > |\sum_t q_{\text{IUL},i}^t| + TC_2$.

Finally, we derive the conclusion:

$$||\theta_{\text{CUL}} - \theta_f||_{X^\top X} = ||\sum_t q_{\text{CUL}}^t||_2 > ||\sum_t q_{\text{CUL}}^t||_2 + TC\sqrt{n}, \tag{49}$$

therefore,

$$||\theta_{\text{CUL}} - \theta_f||_{X^\top X} > ||\theta_{\text{IUL}} - \theta^0||_{X^\top X} + TC\sqrt{n}. \tag{50}$$

In this proposition, we consider a two-task scenario for simplicity. When there are multiple tasks, we can also similarly derive such conclusions:

Corollary C.1. *For multiple tasks, the conclusion of Proposition 2.1 holds. Suppose there are T preceding datasets $\{D^t\}_{t=1}^T$ and a current dataset D^s. If one of the preceding datasets is correlated with D^s, satisfying: $\max_{i \neq j} |\langle x_j, x_i^s \rangle| < k \langle x_i, x_i^s \rangle$, then the previous unlearning results in larger changes of parameters in the unlearning of D^s:*

$$|| \triangle \theta_{CUL}||_{X^\top X} > || \triangle \theta_{IUL}||_{X^\top X} + TC\sqrt{n}. \tag{51}$$

D. Derivation of Equation 9

According to the chain rule, the gradient of Equation 8 with respect to m^t is given by:

$$-\lambda \nabla_{\theta_{\text{ft}}^{t,k+1}} \mathcal{L}_{CE}(\theta^{t-1} - \lambda(\theta_{\text{ft}}^{t,k+1} - \theta^0), D_r) \nabla_{m^t} \theta_{\text{ft}}^{t,k+1} + \mu M \nabla_{m^t} \mathbb{1}_{m^t \geq \eta} \tag{52}$$

For computational efficiency, we approximate the gradient of $\mathcal{L}_{CE}$ on D_r with $\nabla_{\theta^{t-1}} \mathcal{L}_{CE}(\theta^{t-1}, D_r)$. And $\nabla_{m^t} \theta_{\text{ft}}^{t,k+1}$ is derived by substituting into Equation 7:

$$\nabla_{m^t} \theta_{\text{ft}}^{t,k+1} = -\hat{\alpha}^t \odot \mathcal{L}_{ESL}(\theta_{\text{ft}}^{t,k}, D_u^t) \odot \nabla_{m^t} \mathbb{1}_{m^t \geq \eta} \tag{53}$$

For simplicity, we omit $\hat{\alpha}^t$ and approximate $\nabla_{m^t} \mathbb{1}_{m^t \geq \eta}$ using straight-through estimation, which consists in taking this derivative equal to the identity (von Oswald et al., 2021). Then we obtain the gradient of Equation 8 with respect to m^t:

$$-\lambda \mathcal{L}_{ESL}(\theta_{\text{ft}}^{t,k}, D_u^t) \odot \nabla_{\theta^{t-1}} \mathcal{L}_{CE}(\theta^{t-1}, D_r) + \mu M \tag{54}$$

The updating of m^t is given by substracting the gradient:

$$m^t \leftarrow m^t + \lambda \mathcal{L}_{ESL}(\theta_{\text{ft}}^{t,k}, D_u^t) \odot \nabla_{\theta^{t-1}} \mathcal{L}_{CE}(\theta^{t-1}, D_r) - \mu M \tag{55}$$

E. Implementation details

E.1. Optimization

The Adam optimizer is employed for the training of all methods. We employ the learning rate of 3e-5 for the TOFU dataset and 3e-6 for the MUSE News and WHP datasets. Most baselines easily suffer from the issue of over-unlearning while conducting second or third tasks with constant learning rate. Thus, we apply learning rate tuning, decaying the learning rate by 50% after each unlearning task. For experiments in Figure 2 and Figure 8 which require alignment of unlearning performance, learning rates of different methods are tuned to achieve aimed unlearning performance. The hyperparameters s and λ of our method are set as 10 and 0.8. μ is set as the deviation of the unlearning gradients G_u. m^t is initialized with zero vectors in the beginning of each training.

Weight decay ratio is set as 0.01 for all experiments. For TOFU dataset, batch size is set as 32 and models are trained for 5 epochs. For MUSE News dataset, batch size is 6 and number of epochs is 2. For WHP dataset, batch size and epoch number are 6 and 10.

E.2. Tuning of gradient mask threshold η

The threshold η determines the proportion of model parameters that are activated. The threshold η is determined by the percentiles of the underlying vector m^t, and it increases gradually throughout the training process. Specifically, 100% parameters are activated in the beginning of each training, while 30% parameters are activated in the end. With this approach, parameters with smaller m^t values undergo minor fine-tuning in the early stages of training, while parameters with larger m^t values receive more extensive fine-tuning over additional steps. Compared to setting a large threshold η in the beginning of training, this method balances the training of all model parameters, preventing excessive adjustments to a small subset of activated parameters to achieve the unlearning objective.

Calculating the percentiles of m^t can be computationally expensive for models with a large number of parameters since it involves sorting the parameters. To address this, we employ an efficient estimation method. Calculating the maximum and minimum values of a list has a computational complexity of $O(n)$. By identifying the top 15% largest and smallest values, we can mitigate the influence of outliers. The estimated quantiles are then obtained through interpolation between these extreme values after excluding outliers.

E.3. Discussions regarding parameters masking in LLMs unlearning

We argue that an efficient way to preserve the utility of LLM during the unlearning process is to minimize changes to its parameters. In contrast to methods that directly calculate the mask using gradient magnitude, our proposed dynamic gradient sparsity method in Section 3.2 learns the mask values from the data. And the mask is dynamically tuned along with the training process. In this way, each model parameter receives an appropriate training range: parameters with a relatively small impact on the training objective undergo minor adjustments in the early stages, after which they are masked, while parameters with a larger impact are trained more extensively. This approach allows for appropriate training of the model as a whole, avoiding large adjustments to all parameters or drastic changes to a small subset. Besides, the proposed method leverages the underlying vector m to accumulate the influence of data on the mask, thereby avoiding biases introduced by extreme values during a single training iteration.

E.4. Improving Memory Efficiency

Although effective, the iterative approach in Equation 9 requires storing $G_r = \nabla_{\theta^{t-1}} \mathcal{L}_{CE}(\theta^{t-1}, D_r)$ during the fine-tuning process of the t-th task. To reduce memory usage, we propose an alternative method.

To reduce memory overhead during training, G_r can be replaced with a lower-precision approximation, $\hat{G}_r$. For elements with absolute values exceeding a certain threshold, which have a significant impact on the retain set, their signs $+1$ or -1 are preserved, while the remaining elements are set to 0. During the iteration of m^t, calculation of $G_u \odot G_r$ is estimated with:

$$G_u \odot (\hat{G}_r == 0) + a \cdot G_u \odot (\hat{G}_r == 1) - a \cdot G_u \odot (\hat{G}_r == -1), \tag{56}$$

where a is a large positive constant. This method preserves both the signs and magnitude information of G_r while significantly reducing memory consumption.

F. Experimental Setup

F.1. Dataset

TOFU (Maini et al., 2024) dataset consists of information about 200 fictional authors presented in a question-answer pair format. Specific authors' information was designated as unlearning data, while the remaining authors' data were retained to assist in training and testing. Besides, there are also question-answer pairs about *Real Authors* and *World Facts*. We grouped the information of four authors into one unlearning task, resulting in five or more continual unlearning tasks.

MUSE News (Shi et al., 2024b) comprises BBC news articles published after August 2023. This dataset includes four predefined unlearning subsets that serve as continual unlearning tasks. Additionally, it contains a retain set that does not overlap with the unlearning sets.

WHP (Who's Harry Potter) (Eldan & Russinovich, 2023; Shi et al., 2024b) involves the unlearning of original data from the Harry Potter series. In contrast, retain data consists of knowledge related to the books, which aligns with real-world requirements for copyright protection. We treated the original texts of three Harry Potter books as three separate unlearning tasks. To evaluate the performance of algorithms in scenarios involving unrelated continual tasks, we constructed a sequence of continual unlearning tasks by interleaving multiple unlearning tasks from the MUSE News and WHP datasets.

F.2. Metrics

Rouge-L for unlearning (F-Rouge) measures the similarity between the text generated by the model and the correct text. Specifically, given a question or the beginning of the unlearning text, the Rouge-L recall is computed by comparing the output of models with the correct answer or remaining text. A lower Rouge-L value indicates better unlearning performance (Maini et al., 2024; Shi et al., 2024b).

Probability (F-Prob) evaluates the conditional probability $P(\text{answer}|\text{question})$ of the model outputs in the TOFU dataset. To ensure comparability, this probability is normalized by the length of the output text (Maini et al., 2024).

Membership inference attack (MIA) uses the state-of-the-art Min-K% Prob method for LLMs (Shi et al., 2024a) to detect whether data in the unlearning set remains part of the training data for the post-unlearning model. A lower probability of the unlearning set belonging to the training data reflects a more effective unlearning process.

Forget Quality (FQ) quantifies the difference in prediction distributions on the forget set between the post-unlearning model and a retrained model (fine-tuned on retrained data without exposure to the forget data). This metric assesses the degree to which the unlearning algorithm eliminates traces of the forget set in the model (Maini et al., 2024).

R-Rouge and **R-Prob** are also applied to the retain set or holdout set to evaluate the impact of the unlearning algorithm on the overall model utility. Higher values for these metrics indicate less performance degradation.

T-Rouge calculates the Rouge-L recall of completing sentences in the proposed TRAVIS dataset. Testing on inferred training data provides a more accurate assessment of the impact of unlearning algorithms on the pre-existing knowledge of the model. Additionally, TRAVIS is generated without thematic constraints, resulting in a broad coverage of content, which enables a more comprehensive evaluation of performance.

Model Utility (MU) represents the harmonic mean of the above metrics calculated across multiple retain sets in the TOFU dataset (Maini et al., 2024), including TRAVIS dataset. It provides an aggregate measure of the model utility.

F.3. Baseline methods

GA (Yao et al., 2023) unlearns target data by optimizing in the opposite direction of gradient descent. This is achieved by minimizing the negative cross-entropy loss of predicting the next token on the unlearning dataset:

$$\min_{\theta} -\frac{1}{n_u} \sum_{x \in D_u} \sum_{l} -\log \pi_\theta(x_{[l+1]}|x_{[:l]}). \tag{57}$$

NPO (Zhang et al., 2024a) is also a gradient ascent-based approach. Compared to GA, the gradient in NPO is scaled by a regularization term that gradually diminishes over the course of training, alleviating the problem of over-unlearning to some

extent. The loss objective of NPO is:

$$\frac{2}{\beta}\left[\log\left(1+\left(\frac{\sum_l \pi_\theta(x_{[l+1]}|x_{[:l]})}{\sum_l \pi_{\text{ref}}(x_{[l+1]}|x_{[:l]})}\right)^\beta\right)\right] \tag{58}$$

Task Vectors (**TV**) (Ilharco et al., 2023) method manipulates the ability of a pre-trained model to address downstream tasks through fine-tuning and the arithmetic operation of model parameters. When applied to LLM unlearning, the method enables the selective removal of knowledge related to a specific dataset D_u. Specifically, a pre-trained model π_θ is fine-tuned on the unlearning dataset D_u, resulting in a model $\pi_{\theta_{\text{ft}}}$ with enhanced knowledge of D_u. The difference between the fine-tuned parameters and the original parameters, $\theta_{\text{ft}} - \theta$, constitutes the task vector for D_u. By subtracting this task vector from the original parameters θ, a modified model is obtained that has unlearned the knowledge from D_u.

DPO is a preference optimization approach. For a preference dataset, DPO encourages the model to produce one output over the other between a pair. In the unlearning scenario, it encourages the model to produce responses y_w like "I don't know" instead of generating original sensitive text y_l (Zhang et al., 2024a).

SKU (Liu et al., 2024b) is a method based on task vectors. In the fine-tuning process, SKU not only reinforces the target knowledge into the model, but also eliminates knowledge of retain set from the model, incorporating a utility preservation objective. Besides, it includes a robustness objective during the fine-tuning process.

EUL incorporates lightweight unlearning layers into the model to enable efficient unlearning. This method explicitly considers scenarios involving multiple unlearning tasks (Chen & Yang, 2023).

SO-PO and **SO-NPO** introduce a second-order optimization framework for LLM unlearning, leveraging Hessian matrix approximations to address the challenges of applying Newton's method to LLMs (Jia et al., 2024b).

WAGLE identifies model parameters that benefit both unlearning and retaining objectives before training begins, optimizing only these parameters during the unlearning process. This approach is utilized to minimize the impact of unlearning on model utility (Jia et al., 2024a).

When applying these methods in continual unlearning scenarios, the learning rate is tuned progressively to mitigate catastrophic collapse as tasks continue.

G. TRAVIS Dataset

To construct a high-quality text corpus, we adopted the methodology of Membership Inference Attack (MIA) (Yeom et al., 2018) and applied it to two pre-trained language models, Phi-1.5 and Llama2-7b. The approach began with the generation of 5,000 sentences using the models. Subsequently, the MIA technique was employed to identify 250 candidate sentences with the highest probabilities from the generated set (Carlini et al., 2021). After deduplication and filtering based on sentence similarity, the candidate set was further refined. More than 150 high-quality sentences were manually selected to ensure both diversity and authenticity in the dataset.

To enhance the variety of sentence prefixes and enable the models to process a broader range of contextual information, we incorporated a large collection of prefix data sourced from the internet following the work of Carlini et al. (2021). Specifically, we extracted samples from a subset of the Common Crawl dataset, cleaned the data by removing HTML tags and JavaScript code, and performed deduplication, resulting in approximately 50MB of clean text data. From this corpus, we randomly selected 5 to 10 context tokens and employed a Top-n sampling strategy to generate sequences based on the model's probability distribution. The sequences with the highest likelihood were selected, and sentence lengths were capped at a maximum of 1024 tokens to ensure relevance and appropriate length.

The MIA methodology was applied next, using the perplexity of each sentence as a measure of training data likelihood. Perplexity was calculated with the following formula:

$$\mathcal{P} = \exp\left(-\frac{1}{n}\sum_{i=1}^{n}\log f_\theta(x_i \mid x_1, \ldots, x_{i-1})\right), \tag{59}$$

where $x_1, ..., x_n$ denotes a sequence of tokens. Low perplexity indicates that the model holds high confidence in completing the sequence.

However, relying solely on MIA for membership inference introduces potential risks, as the model might assign higher likelihoods to unremarkable or low-quality samples, such as repeated substrings. To address this issue, we introduced several quality control mechanisms to ensure the integrity of the generated text. Specifically, we calculated the zlib entropy of each sample, measured perplexity within a sliding window, and computed the ratio of perplexity before and after lowercasing the text to assess its complexity and information density. These metrics allowed us to filter out low-quality samples and ensure that the selected texts exhibited a high degree of linguistic fluency and semantic richness.

During the selection process, samples were ranked according to the three quality metrics, and the top 100 were chosen. After deduplication, Sentence-BERT was used to compute cosine similarity between sentences, and those with similarity scores above 0.7 were excluded. Cosine similarity was calculated by:

$$\cos(\mathbf{v}_1, \mathbf{v}_2) = \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\|\|\mathbf{v}_2\|}, \tag{60}$$

where $\mathbf{v}_1, \mathbf{v}_2$ denote the embeddings of two sentences. This process yielded approximately 250 diverse and high-quality sentences. In the final manual filtering step, short, highly repetitive, or logically incoherent sentences were discarded, while longer substrings with richer semantics and more varied content were retained. The resulting dataset contains more than 150 high-quality sentences per model, meeting our established quality standards. For dataset construction, the first 70% of each sentence was designated as the prompt, while the remaining 30% served as the ground truth, thus creating the final training dataset.

This dataset offers two significant advantages. First, it accurately evaluates the impact of unlearning algorithms on the original knowledge retained by models. This aligns closely with the definition of efficiency in the context of machine unlearning. Second, the broad thematic coverage of the dataset enables evaluations of model performance across diverse contexts, ranging from everyday conversations to specialized topics. This diversity not only allows the dataset to comprehensively assess the language generation capabilities of the model, but also provides insights into its generalization ability when exposed to different background knowledge. Consequently, the dataset serves as a versatile and reliable foundation for evaluating model performance in a wide range of scenarios.

Table 4. Parts of examples in TRAVIS dataset (Phi Model)

	Phi Model
1	The Concept and Examples Recursion is a powerful programming concept that involves a function calling itself in its definition. This allows the program to be written in a more concise and elegant way, and provides a elegant solution to problems that would otherwise require multiple lines of code. For example: def factorial(n): if n == 0: return 1 return n * factial(n-1) print(factroy(10)) # Prints: 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 3628800
2	Bullying is a pervasive problem that affects many people, especially children. It can have serious long-term effects on mental health, self-esteem, and academic performance.
3	Financial independence is a goal that many individuals aspire to achieve, but the path to attaining it can be challenging. One of the most effective ways to build financial independence early on is through saving.
4	As the world continues to grapple with the effects of climate change, many are turning to creative solutions to help mitigate its impact. From art installations that draw attention to environmental issues to community-based initiatives that promote sustainability, individuals and organizations alike are stepping up to the challenge.
5	Reasons to do exercise regularly: 1.To maintain a healthy weight: Regular exercise helps to burn excess calories and prevent weight gain.

H. More experimental results

H.1. Mixed dataset.

We argue that the higher the data relevance between unlearning tasks, the more severe the cascading degradation phenomenon. To evaluate the performance of our method in scenarios with low data relevance in continual unlearning, we conduct experiments mixing tasks from the MUSE News and WHP datasets. In these experiments, the model is sequentially tasked to unlearn data from the two datasets, and the results are shown in Figure 5. After completing the first MUSE News task, the model proceeds to unlearn the second WHP task, during which all methods exhibit a slight decline in model utility. The decline is less severe compared to unlearning consecutive WHP tasks. Similarly, our method maintains stable model utility when unlearning unrelated tasks sequentially. When moving to the third task—another MUSE News task—baseline

Table 5. Parts of examples in TRAVIS dataset (Llama Model)

	Llama Model
1	The city of Los Angeles also offers a wide range of public services, including healthcare, education, and transportation, all of which are critical in maintaining a high quality of life for city residents.
2	There are a few things you can do to make sure you get the best possible experience from your interactions with the Russian culture and people. 1. Learn as much as you Can:
3	So, if you want to use the power of positive thinking to improve your life, start small and build up gradually. Believe in yourself and your abilities, and don't worry about criticism as it's an integral part of growth.
4	A New York City-based non-profit has been instrumental in fostering greater transparence in the city's supplementary food system. This has involved working with foodstores, restaurants, and community organizations to promote greater understanding and access to healthier foods.
5	The global COVID-19 pandemic has significantly influenced the worldwide popularity of online platforms and digital services.

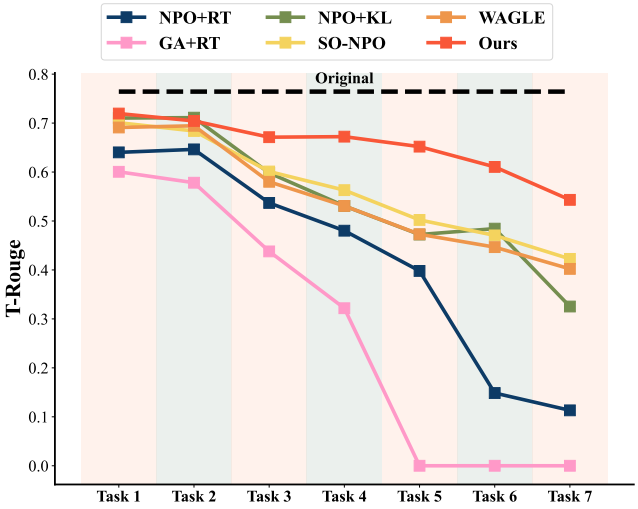


Figure 5. T-Rouge across seven continual unlearning tasks on the mixed dataset of MUSE News and WHP. Different background color represents tasks from different datasets.

methods experienced significant performance degradation due to the cascading degradation effect initiated by the first task. In contrast, our method demonstrates its robustness by effectively mitigating this phenomenon and maintaining model utility.

H.2. Experiments on the TOFU dataset

We present maximum and minimum model utility across 5 unlearning tasks in Figure 6. And the maximum F-Rouge is shown in Table 6, illustrating the performance of the least forgotten task after unlearning. Our proposed method achieves the lowest F-Rouge, outperforming baseline methods in terms of unlearning performance. Figure 6 compares the ability of retaining model utility among different methods. For Max MU, the proposed method attains the highest value (0.5574), followed by WAGLE (0.5486) and SO-NPO (0.5427). Similarly, for Min MU, the proposed method leads with a score of 0.5082, exceeding WAGLE (0.4721) and SO-NPO (0.4503), demonstrating superior retention of utility after unlearning all tasks. These results validate the proposed approach's ability to balance effective unlearning and high model utility.

H.3. Experiments on the MUSE News dataset

MUSE News. Figure 7 (a) illustrates the utility and unlearning results on the MUSE News dataset, which exhibits more severe over-unlearning compared to the TOFU dataset. Specifically, for baseline methods like NPO+RT and GA+RT, the utility drops to zero during the third or fourth unlearning task. This can be attributed to the nature of the dataset, which requires verbatim unlearning of textual data rather than conceptual unlearning. Verbatim unlearning poses a greater challenge and directly disrupts the generative capabilities of the models. Moreover, the initial memorization of the model regarding the news content is relatively shallow; thus, high intensity and continuous unlearning can easily lead to excessive unlearning.

Table 6. Max F-Rouge across five unlearning tasks on TOFU dataset for Llama2 model.

Methods	NPO+RT	GA+RT	NPO+KL	SO-NPO	WAGLE	Ours
Max F-Rouge↓	0.3398	0.3167	0.4538	0.3614	0.4621	0.3067

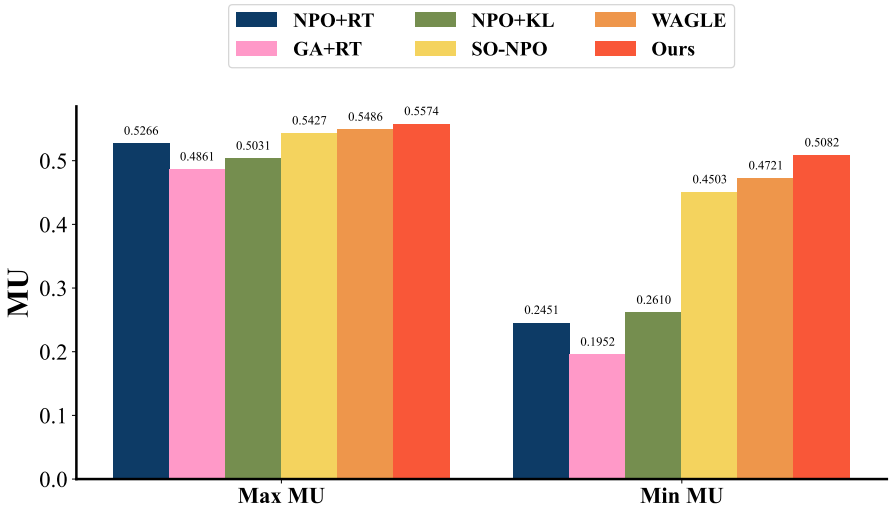
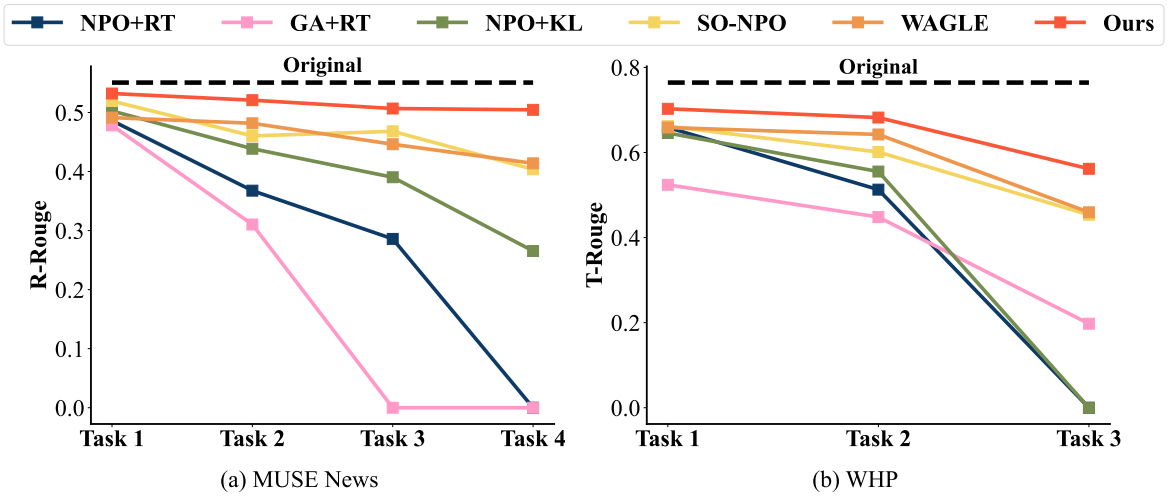


Figure 6. Max/Min Model Utility (MU) across five unlearning tasks on TOFU dataset for Llama2 model.

Our method, leveraging entropic soft labels, adaptively adjusts the unlearning intensity based on the prediction probabilities of the models for each sample. It ensures complete unlearning while dynamically modulating the unlearning intensity and selecting model parameters to avoid redundant unlearning. As a result, our approach maintains model utility effectively, with negligible performance degradation after completing four unlearning tasks.



(a) MUSE News (b) WHP

Figure 7. (a) R-Rouge across four continual unlearning tasks on the MUSE News dataset. (b) T-Rouge across three continual unlearning tasks on the WHP dataset.

H.4. Experiments on the WHP dataset

Figure 7 (b) illustrates the T-Rouge performance of various methods across three continual unlearning tasks on the WHP dataset. T-Rouge measures model utility on the TRAVIS dataset, which includes diverse genres, with higher values of T-Rouge indicating better retention of general knowledge. The ”Original” baseline, shown as the dashed line, represents the model's performance without unlearning. Among the methods, our proposed approach demonstrates superior performance, maintaining the highest T-Rouge scores across all three tasks, indicating minimal degradation in utility. While competing methods like WAGLE and SO-NPO exhibit better retention than NPO+RT and GA+RT, they still show significant drops in performance, especially as tasks progress. As shown in Table 2, our method achieves comparable unlearning performance as other baselines. The consistently higher T-Rouge scores of the proposed method reflect its ability to balance effective unlearning with the retention of general knowledge across diverse tasks, surpassing all baseline methods. This highlights its efficacy in handling challenging continual unlearning scenarios.

H.5. Comparison between continual unlearning and one-time unlearning

To further validate the harmful effect of cascading degradation, we compare the model utility under continual unlearning and one-time unlearning scenarios in Figure 8. For one-time unlearning, all the data for current and previous unlearning tasks is jointly used for unlearning, which is only for experimental purposes and unrealistic in practical applications.

The Rouge-F in these two scenarios are aligned for a fair comparison of model utility. As shown in the figure, continual unlearning results in a significantly more drastic decline in utility, while one-time unlearning results in a mild decrease in utility as the number of tasks increases. This difference indicates the severe damage caused by cascading degradation in utility.

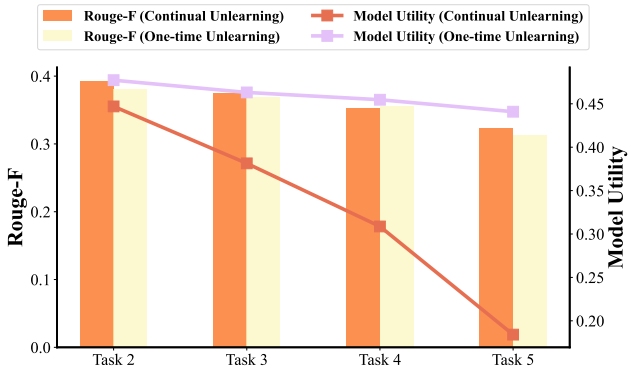


Figure 8. Rouge-F and model utility under two unlearning scenarios.