Avoiding Copyright Infringement via Machine Unlearning

Anonymous ACL submission

Abstract

Pre-trained Large Language Models (LLMs) have demonstrated remarkable capabilities but also pose risks by learning and generating copyrighted material, leading to significant legal and ethical concerns. To address these issues, it is critical for model owners to be able to unlearn copyrighted content at various time steps. We explore the setting of sequential unlearning, where copyrighted content is removed over multiple time steps—a scenario that has not been rigorously addressed. To tackle this challenge, we propose **S**table **S**equential **U**nlearning (**SSU**), a novel unlearning framework for LLMs, designed to have a more stable process to remove copyrighted content from LLMs throughout different time steps using task vectors, by incorporating additional random labeling loss and applying gradient-based weight saliency mapping. Experiments demonstrate that SSU finds a good balance between unlearning efficacy and maintaining model's general knowledge compared to existing baselines. [1]

1 Introduction

Large Language Models (LLMs) (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023) have made significant progress through pre-training on extensive transformer-based architectures and learning from diverse text data (Ouyang et al., 2022; Kojima et al., 2022; Qin et al., 2023; Lewkowycz et al., 2022; Roziere et al., 2023; Lyu et al., 2023; Li et al., 2024). However, LLMs inadvertently incorporate and learn from copyrighted material (Min et al., 2023; Brittain, 2023; Rahman and Santacana, 2023). These issues have led to a lawsuit filed by the New York Times[2] and eight U.S. newspaper publishers[3]. These issues not only pose significant privacy concerns but also raise broader questions regarding responsible AI usage.

In response to these, General Data Protection Regulation of the European Union (Hoofnagle et al., 2019) and the California Consumer Privacy Act (Pardau, 2018) have mandated the *right to be forgotten* (Dang, 2021; Bourtoule et al., 2021). One naive approach is to exclude copyrighted data from training corpus and retrain it from scratch. However, this method is computationally expensive and impractical, as it requires retraining the model each time a copyright violation is identified.

An alternative solution is *machine unlearning* (Cao and Yang, 2015), which removes unwanted knowledge, reconfiguring the model as if it had never learned that data. Recent works proposed practical machine unlearning algorithms for LLMs, discussing the trade-off between privacy and utility (Liu et al., 2024a; Yao et al., 2023; Zhang et al., 2024; Chen and Yang, 2023; Eldan and Russinovich, 2023; Jang et al., 2023; Zhao et al., 2024). However, few have addressed the challenge of *sequentially* unlearning literary copyrighted works. This scenario involves unlearning specific books over time, followed by subsequent unlearning requests. An effective algorithm should be *stable*, meaning it should ensure *unlearning efficacy*—removing unwanted knowledge effectively—while maintaining *locality*, preserving non-targeted knowledge and the model's reasoning ability. Few works have studied this setting, leaving it unclear if existing methods are suitable.

Many previous works have used Gradient Ascent (GA)-based approaches (Zhang et al., 2024; Maini et al., 2024; Zhao et al., 2024; Liu et al., 2024b), often leading to catastrophic collapse — drastically degrading the model's reasoning ability and violating the locality property we desire. This issue is particularly problematic for copyright unlearning, where preserving model performance is crucial. Furthermore, the Task Vector (TV) (Ilharco

[1] Code avilable at https://anonymous.4open.science/r/SSU-E419/README.md.
[2] NYT Complaint, Dec 2023
[3] CNBC, April 2024

et al., 2022) approach fails to achieve a good trade-off among unlearning efficacy, knowledge retention (keeping knowledge of non-unlearned books), and capability retention (maintaining the model's reasoning ability). This failure can degrade the model's overall performance by unintentionally unlearning books that should be retained, leading to a loss of valuable knowledge.

To address these challenges, we propose a **S**table **S**equential **U**nlearning (SSU), marking an initial step toward a better trade-off between effective unlearning and maintaining knowledge and capability retention in sequential settings. SSU is designed to unlearn copyrighted content, thereby avoiding copyright infringement in LLMs. Specifically, we first fine-tune the model with copyrighted books to ensure unlearning efficacy, incorporating a random labeling loss term to enhance stability and applying weight saliency mapping to maintain locality. Then, we negate the learned knowledge during fine-tuning on the original model to obtain a modified model that forgets copyrighted content. Unlike GA-based methods, SSU does not require additional data collection to maintain its performance on other tasks, thereby avoiding the complexity and overhead associated with mitigating catastrophic forgetting. Instead, it leverages internal model mechanisms and loss functions to ensure performance stability.

Our experiments on the Llama3-8B model (AI@Meta, 2024) to sequentially unlearn copyrighted books demonstrate that stable unlearning provides a better trade-off between unlearning efficacy and the retention of model locality compared to baseline methods. This approach alleviates the instability commonly encountered during the unlearning process. Our main contributions are:

- To the best of our knowledge, this is the first work investigating the sequential unlearning of copyrighted literary books to address copyright infringement.
- We systematically evaluate existing algorithms in our sequential unlearning setting and highlight that they either encounter catastrophic collapse or fail to achieve good trade-offs among unlearning efficacy, knowledge retention and capability retention during the unlearning process.
- We propose SSU, a stable unlearning algorithm for sequential setting. Our experiments demonstrate that SSU provides a better trade-off between avoiding copyright infringement and preserving the model's reasoning ability compared to existing methods.

2 Related Work

Machine unlearning was first introduced by Cao and Yang (2015), who proposed using a sharded, isolated, sliced, aggregated (SISA) framework to split the model into smaller sub-models, each learning from a portion of the data. This allows for easier modification of individual sub-models when unlearning is required. There are two main types of unlearning: *Exact Unlearning* and *Approximate Unlearning*. Exact unlearning typically applies to convex settings where all information related to the unwanted data can be completely removed (Ginart et al., 2019; Bourtoule et al., 2021). In contrast, approximate unlearning is used in non-convex settings and requires the output distribution of the unlearned model to be similar to that of a retrained model from scratch (Guo et al., 2020; Sekhari et al., 2021; Liu et al., 2024a; Chien et al., 2022; Pan et al., 2023; Guo et al., 2020). However, neither exact nor approximate unlearning is applicable to LLMs, as it is infeasible to estimate the output distribution of a LLM.

Some studies have specifically addressed unlearning copyrighted content for LLMs. Yao et al. (2023) used a gradient ascent-based approach to unlearn copyrighted contents, while Eldan and Russinovich (2023) explored unlearning the Harry Potter series. However, Shostack (2024) noted that remnants of the Harry Potter books remained in the modified model. Chen and Yang (2023) proposed adding unlearning layers in transformer blocks for sequential data forgetting, but this approach was tested on a smaller model focused on movie reviews in a simulated setting. In contrast, our work targets the sequential unlearning of extensive literary works, a more practical scenario, and addresses the trade-offs between knowledge retention and capability retention more comprehensively.

Furthermore, Chu et al. (2024) proposed a method using softmax regression to prevent large language models from generating copyrighted texts. Fan et al. (2023) studied the instability of some unlearning algorithms for image classification and generation tasks and proposed a gradient-based weight saliency map. Lastly, Maini et al. (2024) and Yao et al. (2024) examined "the right to be forgotten" and provided benchmarks for evaluating the unlearning effectiveness of private data. However, none of these works addressed unlearning copyrighted literary works in a sequential setting or the limitations of existing methods.

3 Preliminaries

3.1 Machine Unlearning for LLMs

Consider the original model and its weights, denoted as θ_o. Machine unlearning involves the problem where, given a dataset $D = \{(x_i, y_i)\}_{i=1}^N$ that θ_o was trained on, we aim to intentionally forget a subset of data, denoted as D_f, to obtain a modified model, denoted as θ_u.

In the context of machine unlearning, we often use a retrained model excluding D_f during pretraining as a gold baseline. However, retraining a model for LLMs is extremely expensive and impractical in real-world settings.

A naive and feasible approach is to perform Gradient Ascent (GA) (Thudi et al., 2022) on D_f. However, previous literature has demonstrated that GA-based methods are prone to catastrophic collapse (Zhang et al., 2024; Liu et al., 2024a; Zhao et al., 2024), even when including gradient descent loss to maintain knowledge retention ability (Liu et al., 2024b). This phenomenon is analogous to catastrophic forgetting in continual learning (McCloskey and Cohen, 1989).

3.2 Task Arithmetic

Unlearning via negating *task vectors* has recently gained attention (Ilharco et al., 2022; Liu et al., 2024b) and has become an important baseline approach for many unlearning tasks. The rationale behind this approach is that by negating the gradient updates of the unwanted data, we can achieve a more localized unlearning algorithm to effectively erase D_f from θ_o.

Specifically, our goal is to forget the dataset D_f. The process involves two stages. First, we perform standard gradient descent to fine-tune θ_o on D_f, resulting in θ_{ft}. Next, we calculate the task vector as the element-wise difference $\theta_{ft} - \theta_o$. We then negate this task vector from θ_o to derive the unlearned model θ_u, expressed as $\theta_u = \theta_o - (\theta_{ft} - \theta_o)$.

3.3 Unlearning with Multiple Time Steps

This section generalizes the unlearning process to multiple time steps. Let D be the original dataset on which the model was trained. Define the set of all data to be forgotten across all time steps T as $D_f = \bigcup_{t=1}^T D_f^t$, where D_f^t represents the subset of data to be forgotten at time step t. Let D_r represent the subset of data to be retained, such that $D_r = D \setminus D_f$. By definition, $D_f \cap D_r = \emptyset$ and $D_f \cup D_r = D$.

At each time step t, we aim to unlearn a specific subset of data D_f^t, resulting in a sequence of modified models $\{\theta^1, \theta^2, \ldots, \theta^T\}$. Here, θ^0 denotes the original model trained on the dataset D, and θ^t denotes the model obtained after unlearning the subsets $D_f^1, D_f^2, \ldots, D_f^t$ sequentially. The objective is to ensure that, after each unlearning step, the model θ^t retains as much general knowledge from D_r as possible while effectively forgetting the data in D_f^t. This sequential unlearning process continues until all specified subsets $D_f^1, D_f^2, \ldots, D_f^T$ have been unlearned.

4 Methods

This section presents SSU, which performs a more stable sequential unlearning and achieves a more balanced trade-off between utility and unlearning efficacy. Unlike the naive Task Vector (TV) approach, which often results in instability due to larger model degradation, SSU leverages task vectors, incorporates additional loss term for ensuring stability and uses a gradient-based weight saliency map to ensure locality. The overall process is shown in Figure 1.

4.1 Learning Stable Task Vectors

First, we present the case of unlearning during the first time step. This means that $t = 1$ and $D_f^1 = D_f$. Following the intuition from task vectors, we first need to fine-tune a model that effectively learns from D_f. To do this, we define $h_\theta(x, y_{y<i}) = \mathbb{P}(y_i | (x, y_{<i}); \theta)$, which is the probability of the token y_i conditioned on the prompt x and the already generated tokens $y_{<i} = [y_1, y_2, ..., y_{i-1}]$. Next, we define the LLM's loss on y as:

$$L(x, y; \theta) := \sum_{i=1}^{|y|} \ell(h_\theta(x, y_{<i}), y_i), \qquad (1)$$

in which l is the cross-entropy loss.

Suppose θ_t is the current LLM through unlearning process. The first goal is to obtain a model that forgets D_f. Specifically, we define our first

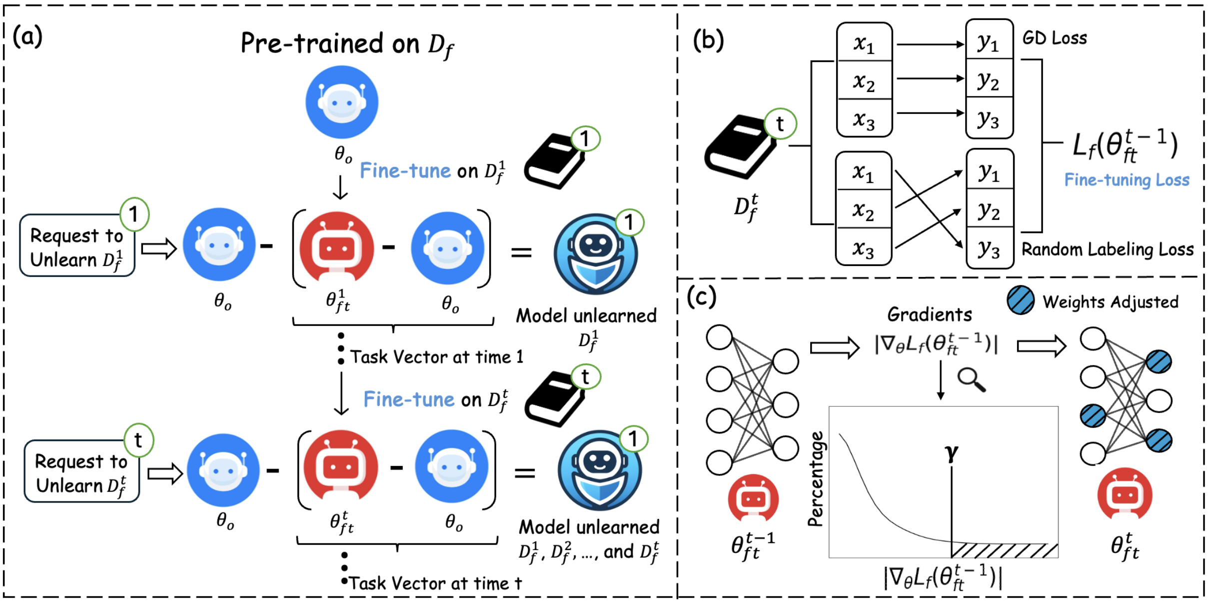


Figure 1: **Overall process of our unlearning framework.** **(a)** At each time step t, an unlearning request is received to forget the dataset D_f^t. The unlearning algorithm involves first fine-tuning θ_{ft}^{t-1} on D_f^t and then subtracting the task vector from the pre-trained model θ_o. **(b)** At each time step t. we compute the gradient loss and random labeling loss to obtain the objective $L_f(\theta_{ft}^{t-1})$ that will be used for fine-tuning. **(c)** We fine-tune θ_{ft}^{t-1} using the objective we obtained in step (b), and only update model weights that are most salient using weight saliency mapping.

gradient descent loss term as:

$$\mathcal{L}_{\text{fgt}} = \sum_{(x_{\text{fgt}}, y_{\text{fgt}}) \in D_f} L(x_{\text{fgt}}, y_{\text{fgt}}, \theta_o). \quad (2)$$

Random Labeling Loss. Inspired by previous works demonstrating that injecting noise during training improves robustness (Miyato et al., 2016; Srivastava et al., 2014; Neelakantan et al., 2015), we propose enhancing the stability of unlearning by introducing data augmentation. Specifically, we randomly mismatch the outputs of D_f with the inputs of D_f. During the first stage of the task vector approach, we include the following loss:

$$\mathcal{L}_{\text{rnd}} := \sum_{(x_{\text{fgt}},) \in D_f} \frac{1}{|D_f|} \sum_{(,y_{\text{rnd}}) \in D_f} L(x_{\text{fgt}}, y_{\text{rnd}}, \theta_t), \quad (3)$$

in which y_{rnd} is any output from D_f and not necessarily corresponds to x_{fgt}.

By incorporating this random labeling loss, we introduce controlled noise into the unlearning process. This helps to prevent "overfitting" and enhance the stability of unlearning. Combining two loss terms, the final objective can be expressed as:

$$L_f(\theta_t) = \epsilon_1 \mathcal{L}_{\text{fgt}} + \epsilon_2 \mathcal{L}_{\text{rnd}}. \quad (4)$$

Weight Saliency. Moreover, to enhance locality of unlearning, we should mitigate the risk of catastrophic collapse during each time step of sequential unlearning. We can achieve this by steering the unlearning process towards specific parts of the model weights that are most relevant to the data to be forgotten. Inspired by this, we use a weight saliency map during the first stage of fine-tuning to further ensure localized unlearning by only adjusting specific weights that are most influenced by the data to be forgotten. The weight saliency map is defined as:

$$m_s = \mathbb{1}(|\nabla_\theta L_f(\theta_t)| \geq \gamma), \quad (5)$$

in which $\mathbb{1}(f \geq \gamma)$ is an element-wise indicator function which outputs one for the i-th element if $f_i \geq \gamma$, and 0 otherwise, and $\nabla_\theta L_f(\theta_t)$ is a gradient vector.

Next, we apply the weight saliency mapping on the parameter that that are most salient to unlearning and have the learned model as at each gradient accumulation step as:

$$\theta_{t+1} = m_s \odot (\Delta\theta + \theta_t) + (1 - m_s) \odot \theta_t, \quad (6)$$

where $\Delta\theta$ indicates model updates. After training for T gradient accumulation steps using Equation 6, we obtain a fine-tuned model θ_{ft}^1. Finally, we obtain our modified model using task vector by negating the knowledge of D_f learned during the fine-tuning process from the original model as:

$$\theta_u^1 = \theta_o - (\theta_{ft}^1 - \theta_o). \quad (7)$$

4.2 Sequential Unlearning

Typically, to sequentially unlearn different data at different time steps, the modified model at previous step is used, and the same unlearning algorithm is applied. However, in SSU, we leverage the fine-tuned model from the previous time step to perform stable sequential unlearning. Specifically, at each time step t, we have the original model $\theta_o = \theta_{ft}^0$ and the previously fine-tuned model θ_{ft}^{t-1}. For each sequential unlearning request, we fine-tune θ_{ft}^{t-1} on D_f^t using the objective described in Equation 6 in Section 4.1 to obtain θ_{ft}^t. Finally, we negate the knowledge learned during fine-tuning to obtain the unlearned model at time step t as:

$$\theta_u^t = \theta_{ft}^t - \theta_{ft}^0.$$

The reason we don't use previously modified model θ_u^{t-1} as the reference model of task vector approach is that we want to avoid accumulated errors that come from each θ_u^{t-1}. If we use θ_u^{t-1} to perform negation difference, each subsequent unlearning step is built upon a potentially degraded model, amplifying any existing errors and making it harder to maintain overall model integrity. Moreover, if we were to reference θ_u^{t-1}, the task vector would reflect not only the new task but also the residual effects of previous tasks and unlearning steps.

5 Experimental Setup

In this section, we present experiments to validate the effectiveness of sequential unlearning of copyrighted books. Our goal is to unlearn copyrighted contents such that the model can avoid generating texts that could potentially infringe copyright laws.

5.1 Settings

To evaluate the effectiveness of sequential unlearning of copyrighted books, we follow the experimental design from (Zhou et al., 2023; Carlini et al., 2022). We unlearn a total of four books, one at each time step. For each book, we split the entire text into chunks of 350 tokens and randomly selected 100 chunks for our experiment. For each chunk, we used the first 200 tokens as the prompt text and a system prompt to ask the model to continue the story, with the following 150 tokens serving as the correct label.

To assess the amount of copyrighted information being leaked, we compared the LLM's completion with the remaining 150 tokens of each chunk from the original book using a greedy decoding strategy. Besides books in D_f, We specifically evaluated performance on three groups of books: (a) books in D_{nor}, (b) books that are not in D_{nor} but are semantically similar any books in D_f (denoted as D_{ss}), and (c) books that are not in D_{nor} and are semantically dissimilar to D_f (denoted as D_{sd}). In subsequent sections, we refer to the performance on books except D_f as knowledge retention. Details about experiment settings are in Appendix A.1.

5.2 Evaluation Metrics

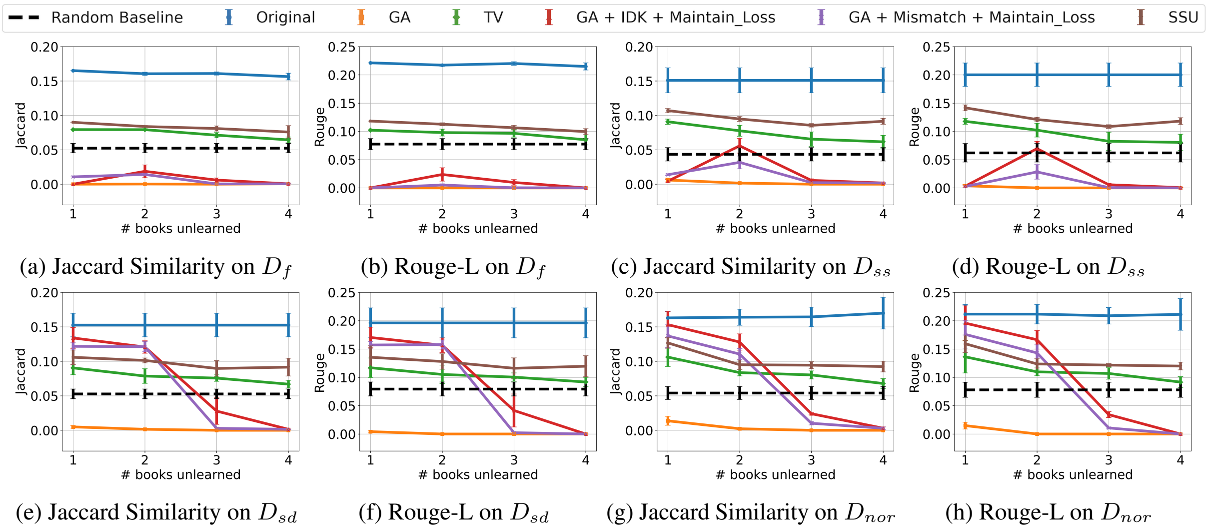
For each prompt, we compared the completion's **Jaccard Similarity** score and **Rouge-L score**. In our experiment, we evaluated these scores on both the books to be forgotten and the books in the retain set D_r, namely D_{nor}, D_{ss}, and D_{sd}. In line with previous unlearning evaluation metrics (Maini et al., 2024; Yao et al., 2024; Chien et al., 2022) and considering that semantic similarity does not indicate copyright infringement, we do not use evaluation metrics that reflect semantic similarity.

In addition to evaluating the model's unlearning effectiveness, we also assessed its performance on general downstream tasks after unlearning, which we refer to as capability retention. The downstream tasks considered include MathQA (Amini et al., 2019), Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2020), and the Graduate-Level Google-Proof Q&A Benchmark (GPQA) (Rein et al., 2023). More details are provided in Appendix A.2.

5.3 Datasets and Models

We used the open-source Llama3-8B (AI@Meta, 2024) language model for our experiments. At time step 1, we unlearned "Harry Potter and the Prisoner of Azkaban" by J.K. Rowling (HP3). Subsequently, we unlearned "Pride and Prejudice" by Jane Austen, "The Adventures of Sherlock Holmes" by Arthur Conan Doyle, and "The Great Gatsby" by F. Scott Fitzgerald at time steps 2, 3, and 4, respectively. These books were chosen due to high Jaccard Similarity and ROUGE-L scores, indicating memorization by the Llama3-8B model.

We initially collected 12 books from Project Gutenberg's "Top 100 EBooks last 30 days" as D_{nor}. At each subsequent time step, the book to be unlearned was removed from D_{nor}. Addition-



(a) Jaccard Similarity on D_f (b) Rouge-L on D_f (c) Jaccard Similarity on D_{ss} (d) Rouge-L on D_{ss}

(e) Jaccard Similarity on D_{sd} (f) Rouge-L on D_{sd} (g) Jaccard Similarity on D_{nor} (h) Rouge-L on D_{nor}

Figure 2: The performance comparison of SSU with baseline methods on four groups of data: (a)(b) – books to forget (D_f); (c)(d) – books that are not in D_{nor} but semantically similar (D_{ss}); (e)(f) – books that are not in D_{nor} but semantically dissimilar (D_{sd}); and (g)(h) – books in D_{nor}. The x-axis of each plots represents different time steps of sequential unlearning. The y-axis shows either the average Jaccard similarity score or the average Rouge-L score. SSU is represented in brown. The black dashed line indicates the random baseline for both Jaccard and Rouge scores. For books to be unlearned, the goal is to approach the random baseline, whereas for other books, the goal is to stay above this baseline.

ally, we included four books semantically similar to HP3 but not in D_{nor} as D_{ss}, and four books not in D_{nor} and semantically dissimilar as D_{sd}. Detailed dataset information is in Appendix A.3.

5.4 Baseline Methods

We compared our approach with state-of-the-art unlearning methods, including GA (Thudi et al., 2022), Task Vectors (Ilharco et al., 2022), and GA with additional loss terms to maintain knowledge (Yao et al., 2023). Specifically, GA with additional loss terms involves using D_{nor} to maintain performance and a random mismatch loss to force LLM to generate random output for unlearned data. The random response could be any labels from D_{nor} or simply the response "I don't know." (IDK) We consider both cases as our baseline methods, referring to them as **GA + Mismatch + Maintain Loss** and **GA + IDK + Maintain Loss**.

Additionally, we derived a random baseline for each book type by mismatching the output of each book with random outputs from other book types and computing Jaccard and ROUGE scores. This approach ensures these random outputs do not infringe copyright, serving as a baseline for determining copyright infringement. A successful unlearning algorithm should aim to match this baseline for D_f while maintaining higher performance on books not D_f. Details are in Appendix A.4.

6 Results

We present experimental results for different unlearning time steps in Figure 2 and Figure 3. See the full results with exact numbers in Appendix B.

6.1 Unlearning Books Sequentially

First, We evaluate the unlearning efficacy of each method on a sequence of books. We task the pre-trained Llama3-8B model to unlearn four books in D_f one at a time. This sequential unlearning setting simulates the situation in which authors of these four books requested model developers to remove their books from the model parameters to protect their copyright.

As shown in Figures 2a and 2b, GA and GA variants have Jaccard and ROUGE scores near zero most of the time. Specifically, the scores are 0 at time steps 1, 3, and 4. At time step 2, the Jaccard score is 0.02 for GA + IDK + Maintain Loss and 0.014 for GA + Mismatch + Maintain Loss, both still well below the random baseline (0.054 for Jaccard and 0.078 for ROUGE). For the naive TV method, the Jaccard score is 0.064 and the ROUGE score is 0.085 at time step 4, which are relatively close to the random baseline. On the other hand, SSU has a Jaccard score of 0.076 and a ROUGE score of 0.099, which are slightly higher than those of the TV method. However, compared to the original model, SSU is already very close (the baseline

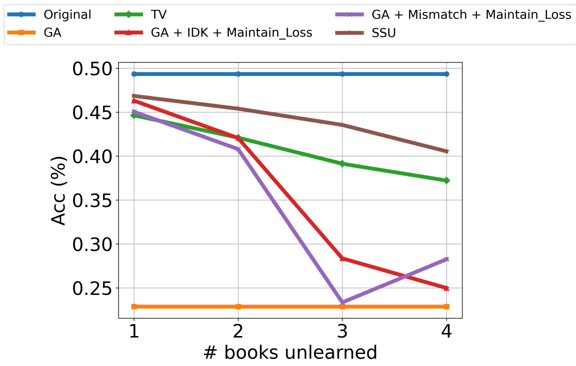


Figure 3: Llama3-8B's Benchmark performance across different unlearing time steps. The x axis is the number of beings unlearned, and the y axis is the average accuracy of MathQA (0-shot), MMLU (0-shot), MMLU (5-shot), and GPQA (0-shot) from main set.

is 28.9% lower) to the random baseline. In conclusion, **SSU effectively minimizes the risk of copyright infringement**.

6.2 Knowledge Retention During Unlearning

This sections studs how unlearning affects the model's knowledge on three groups of books: D_{nor}, D_{ss}, and D_{sd}.

Results for the performance on the additional books collected for GA-based methods D_{nor} are shown in Figures 2g and 2h. Until time step 2, GA + IDK + Maintain Loss and GA + Mismatch + Maintain Loss have high Jaccard and ROUGE scores, which is reasonable as they are intentionally trained on D_{nor} during unlearning process. However, at time steps 3 and 4, their scores drop significantly to near zero due to the unbounded loss function of GA methods, leading to **catastrophic collapse** (Zhang et al., 2024). As a result, the GA-based modified model loses the ability to generate any coherent completions for books in D_{nor} after time step 3. The naive TV method's performance on D_{nor} decreases by 35.85% throughout the time steps. In contrast, SSU preserves knowledge on D_{nor} **36.76%** better than the naive TV and maintains the most stable performance, with only a 26.19% decrease across all time steps.

For books semantically similar to Harry Potter 3, results are shown in Figures 2c and 2d. Except for time step 2, where scores are close to the random baseline, GA-based methods score zero, indicating **over-unlearning** books that are semantically similar to the books to forget. The naive TV method performs better at time step 4, but SSU outperforms all baselines, with a Jaccard score **35%** higher and a ROUGE score **47.5%** higher than TV. At the last time step, SSU's Jaccard is **116.27%** higher, and ROUGE is **93.24%** higher than the baseline.

Performance on books in D_{sd} is shown in Figures 2e and 2f. GA-based methods perform well until time step 2, then catastrophic collapse occur. The naive TV method's performance on D_{sd} decreases throughout the time steps. At time step 4, TV's Jaccard is 26.42% higher, and ROUGE is 16.46% higher than the baseline. SSU still outperforms all baselines, with a Jaccard **35.82%** higher than TV and a ROUGE **30.43%** higher than TV. Additionally, SSU's Jaccard is **74.24%** higher, and ROUGE is **52.90%** higher than the baseline.

In conclusion, **compared to baseline methods, SSU maximally preserves knowledge on books in D_{nor}, D_{ss}, and D_{sd}, making it more stable and maintaining better locality throughout the unlearning process.**

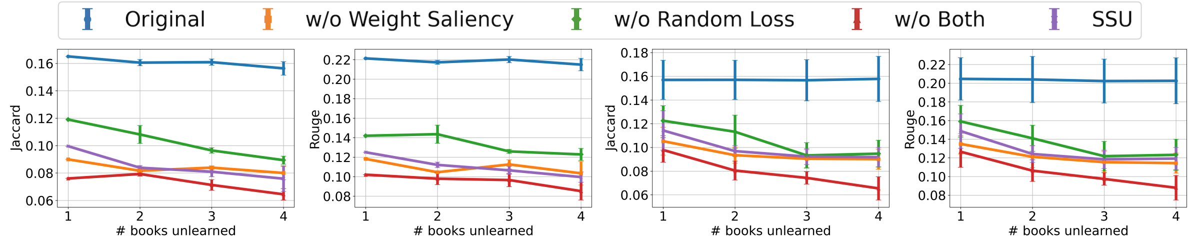
6.3 Capability Retention During Unlearning

We present how sequential unlearning affects model's ability to perform general downstream tasks in Figure 3. Both GA + IDK + Maintain Loss and GA + Mismatch + Maintain Loss suffer from catastrophic collapse at time step 3. Specifically, the GA + IDK + Maintain Loss's average accuracy drops from 0.421 at time step 2 to 0.284 at time step 3, and the GA + Mismatch + Maintain Loss's accuracy drops from 0.408 at time step 2 to 0.233 at time step 3. This indicates a significant loss in reasoning ability.

Meanwhile, SSU results in an average accuracy of 0.436 at time step 3, compared to the TV's average accuracy of 0.391. At time step 4, our model's average accuracy is 0.410, whereas the TV's average accuracy is 0.372. Notably, as shown in Appendix B, at time step 4, TV's MMLU five-shot performance (0.472) is worse than the MMLU zero-shot performance (0.479), indicating that the TV leads the model toward losing its in-context learning ability over time, whereas SSU maintains this capability. **Overall, SSU achieves a better trade-off among unlearning efficacy, knowledge retention, and capability retention comparing to existing baseline methods.**

7 Analysis

In previous section, we demonstrate SSU achieves better trade-off among unlearning efficacy, knowledge retention, and capability retention than exist-



(a) Jaccard Score on D_f (b) Rouge Score on D_f (c) Jaccard Score on D_r (d) Rouge Score on D_r

Figure 4: Ablation study of SSU on each loss terms we introduced during the fine-tuning stage for each time step. For orange line is when we fine-tune without weight saliency map, and green line is when we remove the random labeling loss, and the red line is the case without both components, which is the same as the TV baseline. Lastly, the purple line represents SSU.

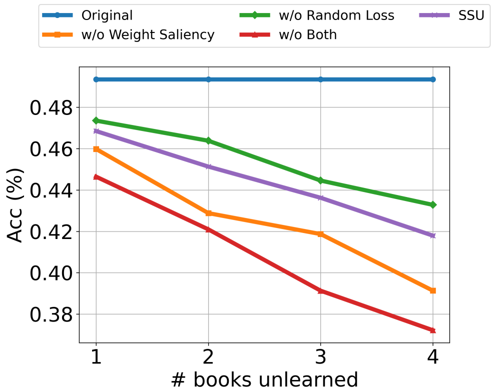


Figure 5: Ablation study of Llama3-8B's Benchmark performance across different unlearing time steps. The x axis is the number of being unlearned, and the y axis is the average accuracy of MathQA (0-shot), MMLU (0-shot), MMLU (5-shot), and GPQA (0-shot main set).

ing baseline methods. In this section, we examine how different components of SSU, including weight saliency maps and random labeling loss, affect the sequential unlearning process. Figure 4 compares unlearning efficacy and knowledge retention and Figure 5 compares capability retention. Note that because D_{nor}, D_{ss}, and D_{sd} are indistinguishable for TV-based methods, we combine all of these books and denote them as D_r.

7.1 How Does Weight Saliency Affect Unlearning?

We study how removing weight saliency during fine-tuning affects overall performance in various aspects of unlearning. As seen in Figure 4, the performance of SSU without weight saliency has a 2.17% lower Jccard score and 5% lower Rouge score on D_r. Moreover, as shown in Figure 5, the benchmark performance of the method without weight saliency decreases much faster at each time step.This suggests that without weight saliency, the risk of catastrophic collapse increases, as the model's reasoning ability deteriorates. **By updating only certain parts of the model weights, weight saliency helps preserve the model's knowledge retention and capability retention, and hence maintains locality**.

7.2 How Does Random Labeling Loss Affect Unlearning?

To understand the role of random labeling loss during sequential unlearning, we conduct an ablation study by removing it from fine-tuning. As seen in Figure 4a and 4b, the unlearning algorithm without random labeling loss has a 17.41% higher Jaccard and 23.30% higher Rouge score on D_f. The performance on D_r remains similar, but the benchmark performance is 1.487% higher without random labeling loss, This indicates that though unlearning algorithm without random labeling loss has a slightly higher benchmark performance, is has a higher risk of copyright infringement. Moreover, the model without random labeling loss shows greater variance across unlearning steps, suggesting that **random labeling loss provides more stable sequential unlearning. This results in a better trade-off among unlearning efficacy, knowledge retention, and capability retention**.

8 Conclusion

In this work, we explore the practical setting of unlearning copyrighted content sequentially from LLMs to mitigate legal and ethical concerns. We propose SSU, which utilizes random labeling loss and gradient-based weight saliency to achieve more stable sequential unlearning. Experiments demonstrate that SSU achieves a better trade-off among unlearning efficacy, knowledge retention, and capability retention compared to existing methods.

9 Limitation

In this work, we primarily use lexical-based evaluation metrics to evaluate the algorithm. However, as Ippolito et al. (2023) notes, measuring verbatim memorization might provide a false sense of privacy. Therefore, we should incorporate methods that can detect the leakage of training data. Membership Inference Attacks (MIAs) (Shokri et al., 2017) offer a promising direction. Nonetheless, current research indicates that the performance of MIAs is near random guessing for pre-trained LLMs in various settings (Duan et al., 2024; Yao et al., 2024). We encourage future research to develop more effective MIAs and apply them to our sequential unlearning setting.

Furthermore, although SSU achieves a better trade-off among unlearning efficacy, knowledge retention, and capability retention compared to state-of-the-art baseline methods, we still observe some loss of knowledge in books that are not meant to be unlearned, and a decrease in the model's reasoning ability. Future work should aim to further minimize the knowledge and capability retention gap between the modified model and the original model to ensure better locality during sequential unlearning.

References

AI@Meta. 2024. Llama 3 model card.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

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

Blake Brittain. 2023. Us copyright office says some ai-assisted works may be copyrighted. *Reuters. March*, 15.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

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

Eli Chien, Chao Pan, and Olgica Milenkovic. 2022. Efficient model updates for approximate unlearning of graph-structured data. In *The Eleventh International Conference on Learning Representations*.

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

Timothy Chu, Zhao Song, and Chiwun Yang. 2024. How to protect copyright data in optimization of large language models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17871–17879.

Quang-Vinh Dang. 2021. Right to be forgotten in the age of machine learning. In *Advances in Digital Science: ICADS 2021*. Springer.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*.

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

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Dennis Wei, Eric Wong, and Sijia Liu. 2023. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. 2020. Certified data removal from machine learning models. In *International Conference on Machine Learning*, pages 3832–3842. PMLR.

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

Chris Jay Hoofnagle, Bart Van Der Sloot, and Frederik Zuiderveen Borgesius. 2019. The european union general data protection regulation: what it is and what it means. *Information & Communications Technology Law*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Daphne Ippolito, Florian Tramer, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher Choquette Choo, and Nicholas Carlini. 2023. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models, 2022. *URL https://arxiv. org/abs/2206.14858*.

Weigeng Li, Neng Zhou, and Xiaodong Qu. 2024. Enhancing eye-tracking performance through multi-task learning transformer. In *International Conference on Human-Computer Interaction*, pages 31–46. Springer.

Zheyuan Liu, Guangyao Dou, Eli Chien, Chunhui Zhang, Yijun Tian, and Ziwei Zhu. 2024a. Breaking the trilemma of privacy, utility, and efficiency via controllable machine unlearning. In *Proceedings of the ACM on Web Conference 2024*, pages 1260–1271.

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

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329.

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

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Sewon Min, Suchin Gururangan, Eric Wallace, Hannaneh Hajishirzi, Noah A Smith, and Luke Zettlemoyer. 2023. Silo language models: Isolating legal risk in a nonparametric datastore. *arXiv preprint arXiv:2308.04430*.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. 2015. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Chao Pan, Eli Chien, and Olgica Milenkovic. 2023. Unlearning graph classifiers with limited data resources. In *Proceedings of the ACM Web Conference 2023*, pages 716–726.

Stuart L Pardau. 2018. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J. Tech. L. & Pol'y*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Noorjahan Rahman and Eduardo Santacana. 2023. Beyond fair use: Legal risk evaluation for training llms on copyrighted text. In *ICML Workshop on Generative AI and Law*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. 2021. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.

Adam Shostack. 2024. The boy who survived: Removing harry potter from an llm is harder than reported. *arXiv preprint arXiv:2403.12082*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. 2022. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 303–319. IEEE.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024. Machine unlearning of pre-trained large language models. *arXiv preprint arXiv:2402.15159*.

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

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

Weixiang Zhao, Yulin Hu, Zhuojun Li, Yang Deng, Yanyan Zhao, Bing Qin, and Tat-Seng Chua. 2024. Towards comprehensive and efficient post safety alignment of large language models via safety patching. *arXiv preprint arXiv:2405.13820*.

Xin Zhou, Yi Lu, Ruotian Ma, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Making harmful behaviors unlearnable for large language models. *arXiv preprint arXiv:2311.02105*.

A Appendix: Experiment Details

A.1 Experiment Settings

To evaluate the effectiveness of sequential unlearning, we conduct experiments on several copyrighted books. Our process involves the following steps:

First, each book is split into 100 chunks of 350 tokens. For each chunk, the initial 200 tokens are used as a prompt, which is fed into the LLM. The remaining 150 tokens serve as the answer or continuation from the original book. This setup allows us to assess how well the model can generate text that follows the given prompt.

In addition to the prompt from the book, we use a system prompt to guide the model in generating the completion. The system prompt is designed to instruct the model to continue the story in a coherent and engaging manner, ensuring consistency with the plot, characters, and writing style of the original book. The complete prompt given to the model is:

> *"Continue the story based on the given context from the book. Generate a coherent and engaging continuation that follows the plot, maintains consistency with the characters, and captures the writing style of the original book."*

For each prompt, the model generates a completion using a greedy decoding strategy by setting the temperature to 0. This method involves selecting the most likely next word at each step, ensuring that the generated text is a plausible continuation of the prompt.

To evaluate the generated completions, we use several metrics, including Jaccard Similarity, ROUGE-L score, and Perplexity. These metrics allow us to compare the LLM's completions with the original text and assess the model's ability to unlearn specific content while retaining its overall language capabilities.

Specifically, we evaluate the scores on the following sets of books:

- Books to be forgotten (D_f)
- Books in D_{nor} (those not to be forgotten but used for maintaining knowledge)
- Books not in D_{nor} but semantically similar
- Books not in D_{nor} and semantically dissimilar

We test books in D_{nor} separately because GA + Mismatch + Maintain Loss and GA + IDK + Maintain Loss learn these books during the unlearning process. In subsequent sections, we refer to the performance on books other than D_f as knowledge retention.

Additionally, we evaluate the model's performance on general downstream tasks to assess its capability retention. The downstream tasks considered include MathQA (0-shot) (Amini et al., 2019), Massive Multitask Language Understanding (MMLU) (0-shot and 5-shots) (Hendrycks et al., 2020), and Graduate-Level Google-Proof Q&A Benchmark (GPQA) (0-shot on main set) (Rein et al., 2023).

A.2 Evaluation Metrics

A.2.1 Jaccard Similarity

Jaccard similarity is a measure of similarity between two sets. It is defined as the size of the intersection divided by the size of the union of the sets. The Jaccard similarity score ranges from 0 to 1, where 0 means no similarity and 1 means complete similarity.

To compute the Jaccard similarity between the LLM's completion (hypothesis text) and the original book (reference text), we follow these steps:

First, we tokenize both texts into sets of words:

$$\text{set}_1 = \text{set of words in the hypothesis text} \tag{8}$$

$$\text{set}_2 = \text{set of words in the reference text} \tag{9}$$

Next, we define the intersection as the set of words common to both texts:

$$\text{Intersection} = \text{set}_1 \cap \text{set}_2 \tag{10}$$

We also define the union as the set of all unique words present in either of the texts:

$$\text{Union} = \text{set}_1 \cup \text{set}_2 \tag{11}$$

The Jaccard similarity is then calculated as the ratio of the size of the intersection to the size of the union:

$$Jaccard\ Similarity = \frac{|\text{Intersection}|}{|\text{Union}|} \tag{12}$$

Here, $|\text{Intersection}|$ represents the number of words that appear in both the hypothesis and reference texts, and $|\text{Union}|$ represents the total number of unique words in both texts combined.

This metric helps us understand the extent of overlap between the LLM's completion and the original book, providing a measure of how similar the two texts are in terms of their word content.

A.2.2 Rouge-L

Recall-Oriented Understudy for Gisting Evaluation (Rouge) measures the longest common subsequence (LCS) between the LLM's completion and original books. In detail, LCS is a sequence that appears in both the completion (hypothesis text) and original book (reference text) in the same order but not necessarily contiguously.

Next, we define the recall as the ratio of the length of the LCS to the total length of the reference text:

$$Recall = \frac{LCS}{\text{length of the reference text}}. \tag{13}$$

We also define the precision as the ratio of the length of the LCS to the total length of the hypothesis text:

$$Precision = \frac{LCS}{\text{length of the hypothesis text}}. \tag{14}$$

Lastly, the Rouge-L score we used in our experiments is defined as:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{15}$$

A.3 Datasets

This section provides detailed information about the books used in the experiment.

A.3.1 Books to Forget

At time step 1, we unlearn the third book of the Harry Potter series (HP3) by J.K. Rowling. Subsequently, we unlearn Pride and Prejudice by Jane Austen, The Adventures of Sherlock Holmes by Arthur Conan Doyle, and The Great Gatsby by F. Scott Fitzgerald at time steps 2, 3, and 4, respectively.

A.3.2 Books not in D_{nor}

Throughout the experiments, we collect four books that are semantically similar to HP3 but not in D_{nor}: Harry Potter 2, Harry Potter 6, The Tales of Beedle the Bard, and Short Stories from Hogwarts of Heroism, Hardship, and Dangerous Hobbies, all written by J.K. Rowling. The last two are stories closely related to the Harry Potter series and hence are also semantically similar.

In addition to the semantically similar books, we collect four books from Project Gutenberg that are semantically dissimilar and not in D_{nor}: Metamorphosis by Franz Kafka, Cranford by Elizabeth Cleghorn Gaskell, A Doll's House: a play by Henrik Ibsen, and Little Women by Louisa May Alcott.

A.3.3 Books in D_{nor}

At time step 1 (unlearning Harry Potter 3), the 12 books collected from Project Gutenberg to be initially used as D_{nor} are: Alice's Adventures in Wonderland by Lewis Carroll, Adventures of Huckleberry Finn by Mark Twain, The Enchanted April by Elizabeth Von Arnim, The Scarlet Letter by Nathaniel Hawthorne, The Great Gatsby by F. Scott Fitzgerald, The Adventures of Sherlock Holmes by Arthur Conan Doyle, Jane Eyre: An Autobiography by Charlotte Brontë, My Life — Volume 1 by Richard Wagner, The Blue Castle: a novel by L.M. Montgomery, Romeo and Juliet by William Shakespeare, Twenty Years After by Alexandre Dumas and Auguste Maquet, and Pride and Prejudice by Jane Austen.

At time step 2, since we are unlearning Pride and Prejudice, we remove Pride and Prejudice from D_{nor}. Similarly, we remove The Adventures of Sherlock Holmes and The Great Gatsby at time steps 3 and 4, respectively.

A.3.4 Preparing the Dataset

For books in D_f and D_r, we split the entire texts into chunks of 400 tokens and format the dataset as QA pairs, in which the first 200 tokens are considered the Question, and the next 200 tokens are considered the Answer. We include all the texts from the book and format them into JSON files.

A.4 Baseline Methods

A.4.1 Unlearning via Gradient Ascent with Other Loss Terms

In this work, we use the method proposed by (Yao et al., 2023) as one of the baseline methods. We first discuss the case of time step 1 and then cover sequential unlearning in section A.4.3.

Specifically, let θ_o denote the original model weight of LLM, θ_t the current LLM through unlearning process, $D_f^1 = D_f$ the dataset representing the book we want to forget, and D_{nor} to a set of book corpora that does not contain the book to be forgotten. Moreover, we define $h_\theta(x, y_{y<i}) = \mathbb{P}(y_i|(x, y_{<i}); \theta)$, which is the probability of the token y_i conditioned on the prompt x and the already generated tokens $y_{<i} = [y_1, y_2, ..., y_{i-1}]$. Next, we define the LLM's loss on y as:

$$L(x, y; \theta) := \sum_{i=1}^{|y|} \ell(h_\theta(x, y_{<i}), y_i) \qquad (16)$$

The GA + Mismatch based method has three loss terms, defined as follows:

$$\mathcal{L}_{\text{fgt}} = -\sum_{(x_{\text{fgt}}, y_{\text{fgt}}) \in D_f} L(x_{\text{fgt}}, y_{\text{fgt}}, \theta_t) \qquad (17)$$

$$\mathcal{L}_{\text{rnd}} := \sum_{(x_{\text{fgt}},) \in D_f} \frac{1}{|Y_{\text{rnd}}|} \sum_{(,y_{\text{rnd}}) \in Y_{\text{rnd}}} L(x_{\text{fgt}}, y_{\text{rnd}}, \theta_t) \qquad (18)$$

$$\phi_\theta = h_\theta(x_{\text{nor}}, y_{\text{nor}<i}) \qquad (19)$$

$$\mathcal{L}_{\text{nor}} := \sum_{(x_{\text{nor}}, y_{\text{nor}}) \in D_{\text{nor}}} \sum_{i=1}^{|y_{\text{nor}}|} \text{KL}(\phi_{\theta_o} \parallel \phi_{\theta_t}). \qquad (20)$$

in which Y_{rnd} is a set of responses irrelevant to responses of D_f.

Lastly, the GA approach is trying to minimize the following loss to obtain the unlearned model:

$$L = \epsilon_1 \mathcal{L}_{\text{fgt}} + \epsilon_2 \mathcal{L}_{\text{rnd}} + \epsilon_3 \mathcal{L}_{\text{nor}} \qquad (21)$$

$$\theta_{t+1} \leftarrow \theta_t - \nabla L.$$

in which $\mathcal{L}_{\text{fgt}}$ is a gradient ascent loss on D_f, which tries to make the model perform poorly on the D_f. Next, $\mathcal{L}_{\text{rnd}}$ tries to randomly mismatch the labels from non-relevant dataset to the inputs of the dataset we want to forget. Lastly, $\mathcal{L}_{\text{nor}}$ tries to maintain the performance on the normal dataset. In the end, after T gradient accumulation steps, we obtain the unlearned model θ_u^1.

In our work, we consider two different settings for the Y_{rnd} in the loss term $\mathcal{L}_{\text{rnd}}$. First case is when we consider all the responses in D_{nor} as Y_{rnd}, and we refer this as GA + Mismatch + Maintain Loss. The second setting is we consider the answer "I don't know" as Y_{rnd}, and we refer the second setting as GA + IDK + Maintain Loss.

A.4.2 Unlearning via Task Vector

We also use the task vector method as one of the baseline approaches, which typically involves a two-stage process. Considering the case of $t = 1$, we denote θ_o as the original model weights. We intentionally fine-tune the model on D_f to obtain θ_{ft}^1. This fine-tuning process is defined as follows:

$$\mathcal{L}_{\text{fgt}} = \sum_{(x_{\text{fgt}}, y_{\text{fgt}}) \in D_f} L(x_{\text{fgt}}, y_{\text{fgt}}, \theta_t) \qquad (22)$$

$$\theta_{t+1} \leftarrow \theta_t - \epsilon \nabla_{\theta_t} \mathcal{L}_{\text{fgt}} \qquad (23)$$

Next, we define the task vector τ as the element-wise difference between θ_{ft} and θ_o:

$$\tau = \theta_{ft}^1 - \theta_o \tag{24}$$

Finally, the unlearned model θ_u at time step t is obtained by:

$$\theta_u^1 = \theta_o - \tau \tag{25}$$

The general intuition behind this method is to first obtain a model that is specialized in the dataset we aim to forget. The task vector τ represents the changes in weights required to acquire this specific knowledge. By subtracting these "knowledge" weights from the original model, we effectively unlearn the targeted information.

A.4.3 Sequential Unlearning

For GA, GA + Mismatch + Maintain Loss, and GA + IDK + Maintain Loss, we apply the same algorithm described in Appendix A.4.1 to the previously unlearned model θ_u^{t-1} at each time step t to perform sequential unlearning. For the TV approach, we use the previously fine-tuned model weights and follow the method described in section 4.1 to perform sequential unlearning.

A.5 Implementation Details

The experiments are conducted on four RTX A6000 GPUs. For all unlearning algorithms, at each time step, we perform 200 gradient accumulation steps. The batch size is set to 4, and the learning rate is maintained at 0.001 throughout the experiment. Additionally, we set γ to the mean of the gradient vector $\nabla_\theta L_f(\theta_t)$.

B Appendix: Complete Experiment Results

In this section, we present our experimental results numerically. Table 1 shows the results of unlearning "Harry Potter and the Prisoner of Azkaban" (HP3) at the first time step. Table 2 provides the results when we continuously unlearn "Pride and Prejudice." Table 3 displays the results of further unlearning "The Adventures of Sherlock Holmes," and Table 4 presents the results of unlearning "The Great Gatsby" at the final time step. As described in Appendix A.3, we adjust D_{nor} at each subsequent time step, resulting in different numbers for the original model. For each set of books, we present the average score.

At time step 2, the 5-shot performance of GA + IDK + Maintain Loss is lower than the 0-shot performance, indicating that the model has deteriorated in its ability to follow instructions and perform in-context learning. At time step 3, catastrophic collapse occurs for both GA-based methods. Moreover, SSU consistently performs better in terms of achieving a better trade-off among unlearning efficacy, the model's performance on D_{nor}, D_{ss}, D_{sd}, and benchmark performance across all time steps compared to baseline methods.

	$\mathbf{D_f}$		$\mathbf{D_{nor}}$		$\mathbf{D_{ss}}$		$\mathbf{D_{sd}}$		**Benchmark**				
	Jaccard	Rouge	Jaccard	Rouge	Jaccard	Rouge	Jaccard	Rouge	MathQA	MMLU (0-shot)	MMLU (5-shot)	GPQA	Avg
Original	0.165	0.221	0.164	0.212	0.151	0.200	0.153	0.196	0.402	0.618	0.648	0.306	0.494
GA	0	0	0.013	0.016	0.006	0.004	0.005	0.004	0.188	0.247	0.247	0.234	0.229
Task Vector	0.076	0.102	0.106	0.137	0.091	0.118	0.090	0.117	0.359	0.573	0.603	0.250	0.446
GA + IDK + Maintain Loss	0	0	0.153	0.197	0.004	0.003	0.134	0.170	0.381	0.587	0.617	0.268	0.463
GA + Mismatch + Maintain Loss	0.011	0	0.135	0.180	0.014	0.002	0.122	0.157	0.350	0.566	0.603	0.284	0.451
SSU	0.090	0.125	0.126	0.162	0.107	0.142	0.106	0.135	384	0.590	0.614	0.286	0.469

Table 1: Overall results of our proposed method compared with several baselines at time step 1. D_f consists of HP3, while D_{nor} includes the books collected for GA-based methods. D_{ss} comprises books that are not in D_{nor} but are semantically similar to HP3, and D_{sd} includes books that are not in D_{nor} and are semantically dissimilar. For each type of book, we present the average score. For benchmark performance, we present the accuracy of MathQA, MMLU under 0-shot and 5-shot settings, and GPQA's main set under the 0-shot setting.

	$\mathbf{D_f}$		$\mathbf{D_{nor}}$		$\mathbf{D_{ss}}$		$\mathbf{D_{sd}}$		**Benchmark**				
	Jaccard	Rouge	Jaccard	Rouge	Jaccard	Rouge	Jaccard	Rouge	MathQA	MMLU (0-shot)	MMLU (5-shot)	GPQA	Avg
Original	0.161	0.217	0.164	0.212	0.151	0.200	0.153	0.196	0.402	0.618	0.648	0.306	0.494
GA	0	0	0.002	0	0.006	0	0.001	0.004	0.187	0.246	0.247	0.234	0.228
Task Vector	0.079	0.098	0.084	0.109	0.078	0.102	0.079	0.105	0.338	0.541	0.552	0.253	0.421
GA + IDK + Maintain Loss	0.019	0.024	0.128	0.166	0.056	0.069	0.121	0.156	0.366	0.541	0.519	0.257	0.421
GA + Mismatch + Maintain Loss	0.014	0.010	0.137	0.143	0.032	0.028	0.121	0.158	0.344	0.477	0.525	0.285	0.408
SSU	0.084	0.112	0.095	0.124	0.095	0.121	0.101	0.128	0.362	0.573	0.594	0.288	0.454

Table 2: Overall results of our proposed method compared with several baselines at time step 2. D_f consists of HP3 and Pride and Prejudice, while D_{nor} includes the books collected for GA-based methods and adjusted accordingly. D_{ss} comprises books that are not in D_{nor} but are semantically similar to HP3, and D_{sd} includes books that are not in D_{nor} and are semantically dissimilar. For each type of book, we present the average score. For benchmark performance, we present the accuracy of MathQA, MMLU under 0-shot and 5-shot settings, and GPQA's main set under the 0-shot setting.

	$\mathbf{D_f}$		$\mathbf{D_{nor}}$		$\mathbf{D_{ss}}$		$\mathbf{D_{sd}}$		**Benchmark**				
	Jaccard	Rouge	Jaccard	Rouge	Jaccard	Rouge	Jaccard	Rouge	MathQA	MMLU (0-shot)	MMLU (5-shot)	GPQA	Avg
Original	0.161	0.220	0.164	0.209	0.151	0.200	0.153	0.196	0.402	0.618	0.648	0.306	0.494
GA	0	0	0	0	0	0	0	0	0.187	0.247	0.247	0.234	0.229
Task Vector	0.071	0.097	0.080	0.107	0.066	0.102	0.076	0.100	0.321	0.507	0.494	0.243	0.391
GA + IDK + Maintain Loss	0.006	0.010	0.024	0.034	0.006	0.069	0.028	0.041	0.291	0.324	0.252	0.268	0.284
GA + Mismatch + Maintain Loss	0	0	0.010	0.011	0.002	0.028	0.003	0.002	0.201	0.229	0.243	0.261	0.233
SSU	0.081	0.106	0.094	0.122	0.086	0.121	0.090	0.116	0.343	0.543	0.554	0.3013	0.436

Table 3: Overall results of our proposed method compared with several baselines at time step 3. D_f consists of HP3, Pride and Prejudice, and Adventures of Sherlock Holmes, while D_{nor} includes the books collected for GA-based methods and adjusted accordingly. D_{ss} comprises books that are not in D_{nor} but are semantically similar to HP3, and D_{sd} includes books that are not in D_{nor} and are semantically dissimilar. For each type of book, we present the average score. For benchmark performance, we present the accuracy of MathQA, MMLU under 0-shot and 5-shot settings, and GPQA's main set under the 0-shot setting.

	$\mathbf{D_f}$		$\mathbf{D_{nor}}$		$\mathbf{D_{ss}}$		$\mathbf{D_{sd}}$		**Benchmark**				
	Jaccard	Rouge	Jaccard	Rouge	Jaccard	Rouge	Jaccard	Rouge	MathQA	MMLU (0-shot)	MMLU (5-shot)	GPQA	Avg
Original	0.156	0.215	0.170	0.211	0.151	0.200	0.153	0.196	0.402	0.618	0.648	0.306	0.494
GA	0.001	0	0	0	0	0	0	0	0.188	0.247	0.247	0.234	0.229
Task Vector	0.064	0.085	0.068	0.916	0.067	0.805	0.067	0.092	0.303	0.479	0.472	0.234	0.372
GA + IDK + Maintain Loss	0	0	0	0.034	0.006	0	0.001	0	0.211	0.289	0.256	0.243	0.250
GA + Mismatch + Maintain Loss	0	0	0	0.011	0.002	0	0.001	0	0.266	0.276	0.329	0.259	0.283
SSU	0.076	0.099	0.093	0.120	0.091	0.118	0.092	0.120	0.328	0.512	0.532	0.270	0.410

Table 4: Overall results of our proposed method compared with several baselines at time step 4. D_f consists of HP3, Pride and Prejudice, Adventures of Sherlock Holmes, and the Great Gatsby, while D_{nor} includes the books collected for GA-based methods and adjusted accordingly. D_{ss} comprises books that are not in D_{nor} but are semantically similar to HP3, and D_{sd} includes books that are not in D_{nor} and are semantically dissimilar. For each type of book, we present the average score. For benchmark performance, we present the accuracy of MathQA, MMLU under 0-shot and 5-shot settings, and GPQA's main set under the 0-shot setting.