SAMPLE-WISE EARLY STOPPING FOR LLM UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) unlearning aims to remove the influence of designated training data while preserving model utility, with applications to privacy protection and regulatory compliance. However, existing methods typically use a fixed training budget, overlooking differences in unlearning difficulty across samples. Consequently, some samples remain insufficiently unlearned, whereas others undergo excessive unlearning, resulting in redundant computation or degraded model utility. To address this challenge, we propose Sample-Wise Early Stopping (SWES), an efficient framework for adaptive sample-wise LLM unlearning. SWES measures each sample's unlearning progress by comparing its length-normalized negative log-likelihood before and during unlearning. It combines temporal smoothing with threshold-based control to dynamically suspend or resume each sample's participation in backpropagation, reducing unnecessary updates and focusing subsequent training on samples that have yet to reach the unlearning target. As a plug-and-play framework, SWES can be integrated into a range of gradient-based unlearning methods. Experiments on TOFU and MUSE demonstrate that SWES improves the trade-off between unlearning effectiveness and model utility in multiple settings, while reducing cumulative sample participation in backpropagation by 15.9%–64.6% and achieving training speedups of $1.15\times$–$2.34\times$.

1 INTRODUCTION

As large language models (LLMs) are increasingly adopted for information retrieval (Zhai, 2024; Sun et al., 2023), content generation (Li et al., 2024), and human–AI interaction (Zhao et al., 2024), concerns about their safety and compliance have grown. Studies have shown that LLMs may memorize and reproduce training data, posing risks of privacy leakage (Carlini et al., 2021; Nasr et al., 2025) and copyright infringement (Karamolegkou et al., 2023; Lee et al., 2023), and may also generate harmful content (Mazeika et al., 2024). These concerns have created a need for models to forget specific knowledge (Dou et al., 2025; Shi et al., 2024). However, removing the influence of targeted training data through retraining is often costly (Liu et al., 2025). LLM unlearning therefore aims to remove the influence of such data without full retraining, while preserving the model's other knowledge and capabilities as much as possible (Yuan et al., 2025b; Tian et al., 2024).

Previous studies have shown that unlearning difficulty varies substantially across samples (Krishnan et al., 2025; Feng et al., 2025), leading to uneven forgetting progress during training (Shao et al., 2026). To illustrate this issue, Figure 1 (left) presents the forgetting trajectories, showing that samples require different numbers of training epochs to reach a predefined unlearning level. Figure 1 (right) further shows the distribution of first target-reaching epochs: some samples reach the target early, whereas others require more epochs or fail to reach it within the training budget. Consequently, assigning the same training budget to all samples may leave some insufficiently unlearned, while those that have already reached the target continue to contribute to parameter updates, incurring redundant computation and potentially degrading model utility. This motivates the following research question: **Can we adaptively control each sample's participation in backpropagation based on its suppression progress to reduce unnecessary computation while balancing unlearning effectiveness and model utility?**

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

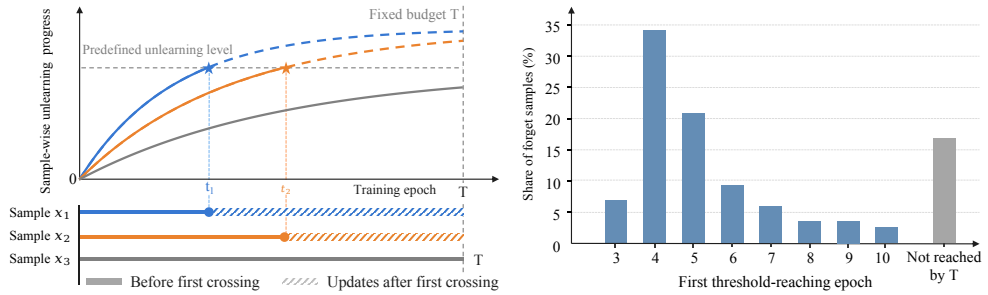


Figure 1: Motivation for sample-wise early stopping in LLM unlearning. **Left:** Forget samples reach the target at different epochs, while some remain below it throughout budget T. Solid and dashed segments indicate updates before and after first target attainment, respectively; the latter may be redundant. **Right:** Distribution of first target-reaching epochs; "Not reached by T" denotes samples that do not reach the target within the budget.

Prior work has addressed differences in forgetting progress across samples through objective design, loss reweighting, and sample selection. NPO (Zhang et al., 2024) and SimNPO (Fan et al., 2025) regulate update dynamics through distinct unlearning objectives, while SatImp (Yang et al., 2025) adjusts contributions to optimization based on saturation and importance. BalDRO (Shao et al., 2026) prioritizes hard-to-unlearn samples through high-loss sample selection or continuous reweighting. These methods focus on how to allocate optimization effort across samples, whereas we investigate when further unlearning updates for individual samples can be suspended. Early stopping provides a starting point: MASC (Gennaro et al., 2026) determines when to terminate unlearning at the model level, while IES (Yuan et al., 2025a; 2026b), developed for conventional training, uses second-order differences in per-sample loss values to assess learning status and dynamically exclude mastered samples from subsequent backpropagation. However, global early stopping cannot readily accommodate differences in forgetting progress across samples, and the second-order difference criterion used by IES captures changes in loss without directly measuring the extent to which the likelihood of a target response has been suppressed relative to the pre-unlearning model. Therefore, sample-level early stopping for LLM unlearning requires a stopping criterion aligned with the unlearning objective.

To this end, we propose Sample-Wise Early Stopping (SWES), an adaptive sample-wise early stopping framework for LLM unlearning. SWES measures each sample's relative likelihood suppression by comparing its length-normalized negative log-likelihood under the current and pre-unlearning models. It combines temporal smoothing with thresholding to dynamically determine whether each sample participates in subsequent backpropagation. When the smoothed suppression signal reaches a predefined threshold, SWES suspends backpropagation for that sample while continuing to monitor its state, and resumes training on it if the signal falls below the threshold. The framework preserves the base method's per-sample loss formulation and the weights of its objective terms, controlling unlearning by dynamically adjusting sample participation in training. It can be integrated with different gradient-based unlearning methods to reduce unnecessary updates while balancing unlearning effectiveness and model utility.

We evaluate SWES on the TOFU and MUSE benchmarks by integrating it into NPO, SimNPO, and WGA. Experimental results show that SWES reduces the total number of times samples from the forget and retain sets participate in backpropagation by 15.9%–64.6% and achieves a 1.15–2.34$\times$ training speedup after accounting for additional overhead. It also improves the trade-off between unlearning effectiveness and model utility in multiple settings. Our main contributions are as follows:

- We propose SWES, a sample-wise early stopping framework for LLM unlearning that dynamically adjusts sample participation in training to reduce unnecessary backpropagation computation, while preserving the base method's per-sample loss formulation and the weights of its objective terms.
- We design an unlearning criterion based on likelihood suppression relative to a reference model. It quantifies each sample's relative suppression using the difference in length-

normalized negative log-likelihood between the current and pre-unlearning models, and uses temporal smoothing to guide sample-wise stopping and reactivation.

- We evaluate SWES on TOFU and MUSE in terms of computational efficiency and the trade-off between unlearning effectiveness and model utility. We further analyze the roles of sample-wise control and the proposed criterion through comparisons with global early stopping under comparable backpropagation budgets and with alternative stopping criteria.

2 Related Work

LLM Unlearning. Existing LLM unlearning methods mainly include model editing (Li et al., 2026; Shen et al., 2025; Yu et al., 2025), gradient-based optimization (Yao et al., 2024; Zhang et al., 2024; Fan et al., 2025; Cha et al., 2025), and in-context interventions (Pawelczyk et al., 2024; Wang et al., 2025b; Bhaila et al., 2025). We focus on gradient-based unlearning methods that are compatible with existing fine-tuning pipelines. These methods fall into two categories: one constructs refusal or alternative responses to guide the model toward specified outputs (Wang et al., 2025c; Mekala et al., 2025; Eldan & Russinovich, 2023), while the other directly reduces the probability of generating target responses. The latter includes gradient ascent (GA), which maximizes the negative log-likelihood of target responses; gradient difference (GD), which incorporates a retention objective; and NPO (Zhang et al., 2024), SimNPO (Fan et al., 2025), and WGA (Wang et al., 2025a), which regulate unlearning updates through different mechanisms. Recent work has further examined differences in forgetting across samples: SatImp (Yang et al., 2025) combines saturation and importance for loss reweighting, MRD (Feng et al., 2025) performs weighted sampling based on unlearning difficulty, and BalDRO (Shao et al., 2026) builds on distributionally robust optimization to coordinate forgetting progress through hard sample selection or continuous reweighting. SWES dynamically determines each sample's subsequent participation in training based on whether it has reached a predefined level of relative likelihood suppression, while preserving the base method's per-sample loss formulation and the weights of its objective terms.

Early Stopping. Early stopping is a widely used training strategy that reduces computational overhead and mitigates overfitting by terminating training at an appropriate point (Prechelt, 2012; Caruana et al., 2000; Yuan et al., 2026a; 2024; Yao et al., 2024). Traditional early stopping typically determines a single stopping point based on validation performance (Prechelt, 2012; Caruana et al., 2000). IES (Yuan et al., 2025a; 2026b) extends early stopping to the sample level, using second-order differences in per-sample loss values to identify mastered samples, exclude them from subsequent backpropagation, and allow excluded samples to rejoin training. Recent studies have further applied early stopping to LLM fine-tuning and unlearning: an early stopping strategy based on n-gram memorization scores can reduce verbatim memorization during fine-tuning (Slack & Moubayed, 2025), while MASC (Gennaro et al., 2026) determines when to terminate unlearning using an online stopping condition. SWES focuses on sample-wise early stopping during unlearning, using a criterion based on likelihood suppression relative to a reference model to accommodate differences in forgetting progress across samples. Unlike the second-order difference criterion used by IES, this signal measures the decrease in the likelihood of a target response relative to the pre-unlearning model rather than the stability of changes in loss.

3 Methodology

As shown in Figure 2, SWES consists of three components. First, it uses the reference-relative likelihood suppression (RLS) metric to quantify each sample's suppression progress relative to the reference model. Next, a sample-wise early stopping controller dynamically determines each sample's participation in training based on the smoothed signal. Finally, SWES constructs forget batches from active samples and, when the base method uses retain data, pairs each with an equally sized retain batch. This batch construction enables integration with different gradient-based unlearning methods.

Preliminaries. Let $\mathcal{D}_f = \{(x_i, y_i)\}_{i=1}^{N_f}$ denote the forget set, where x_i is the input and $y_i = (y_{i,1}, \ldots, y_{i,|y_i|})$ is the corresponding response targeted for unlearning. Let $\mathcal{D}_r$ denote the retain set, and let θ_0 and θ_t denote the model parameters before unlearning and at the end of epoch t, respectively.

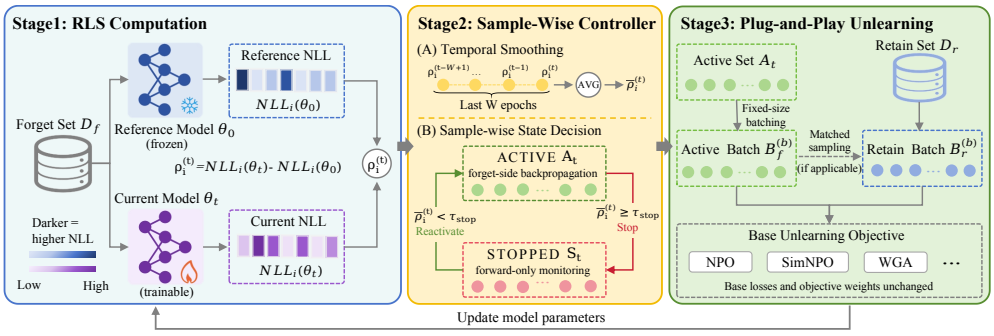


Figure 2: Overview of the SWES framework. (1) RLS computation measures sample-wise likelihood suppression by comparing length-normalized NLLs under the original and current models. (2) The sample-wise controller uses temporally smoothed RLS for threshold-based stopping and reactivation. All forget samples are evaluated without gradients after each epoch. (3) Plug-and-play unlearning trains on active forget samples, with matched retain batches when required.

The length-normalized negative log-likelihood (NLL) (Bengio et al., 2003) of sample i is defined as

$$\ell_i(\theta) = -\frac{1}{|y_i|} \sum_{j=1}^{|y_i|} \log p_\theta \left(y_{i,j} \mid x_i, y_{i,<j} \right), \tag{1}$$

where $|y_i|$ is the response length in tokens and $y_{i,<j}$ denotes the response prefix preceding token j.

3.1 Reference-Relative Likelihood Suppression

Because initial likelihoods under the pre-unlearning model vary across samples, the same absolute NLL does not imply the same degree of relative suppression. Using the pre-unlearning model θ_0 as the reference, we define the reference-relative likelihood suppression (RLS) metric for sample i at the end of epoch t as $\rho_i^{(t)} = \ell_i(\theta_t) - \ell_i(\theta_0)$. Equivalently,

$$\rho_i^{(t)} = \frac{1}{|y_i|} \log \frac{p_{\theta_0}(y_i \mid x_i)}{p_{\theta_t}(y_i \mid x_i)}. \tag{2}$$

At the start of unlearning, $\rho_i^{(0)} = 0$. Larger RLS values indicate greater suppression of the target response's likelihood relative to the pre-unlearning model. The reference NLL $\ell_i(\theta_0)$ can be precomputed and cached, so subsequent evaluations require only the NLL under the current model.

For a suppression threshold $\tau_{\text{stop}} > 0$, if $\rho_i^{(t)} \geq \tau_{\text{stop}}$, then

$$\frac{p_{\theta_t}(y_i \mid x_i)^{1/|y_i|}}{p_{\theta_0}(y_i \mid x_i)^{1/|y_i|}} = e^{-\rho_i^{(t)}} \leq e^{-\tau_{\text{stop}}}. \tag{3}$$

Thus, the geometric mean token likelihood of the target response is at most $e^{-\tau_{\text{stop}}}$ times its pre-unlearning value. This gives a common threshold a consistent interpretation in terms of relative suppression. This relation applies to an individual RLS observation, whereas the actual stopping decision is based on the smoothed signal described in Section 3.2.

NPO (Zhang et al., 2024) incorporates reference-model likelihood ratios into its unlearning objective, while SimNPO (Fan et al., 2025) incorporates length-normalized log-likelihoods. SWES uses this length-normalized, reference-relative signal to control sample participation in training rather than to construct an additional loss. RLS quantifies relative likelihood suppression, but does not establish that a sample has been completely unlearned.

3.2 Sample-Wise Early-Stopping Controller

SWES maintains a training participation state for each sample in the forget set. At the end of each epoch, it evaluates the entire forget set under the current model θ_t using forward passes without gradient computation and updates each sample's RLS. To reduce short-term fluctuations, we compute

a moving average of the RLS observations from the most recent W epochs:

$$\bar{\rho}_i^{(t)} = \frac{1}{W} \sum_{k=t-W+1}^{t} \rho_i^{(k)}, \qquad t \geq W. \tag{4}$$

Let $s_i^{(t)} \in \{\text{Active}, \text{Stopped}\}$ denote the state of sample i, initialized as $s_i^{(0)} = \text{Active}$. The state is updated at the end of epoch t and determines whether the sample participates in backpropagation during epoch $t+1$:

$$s_i^{(t)} = \begin{cases} \text{Stopped}, & t \geq W \text{ and } \bar{\rho}_i^{(t)} \geq \tau_{\text{stop}}, \\ \text{Active}, & \text{otherwise.} \end{cases} \tag{5}$$

Before W observations have been collected, all samples remain active. Thereafter, samples whose smoothed RLS is at or above the threshold stop participating in backpropagation, while those below the threshold continue or resume training. Updates to shared model parameters may change the suppression level of stopped samples. We therefore continue to evaluate these samples at the end of each epoch so that they can be reactivated. Based on the updated states, we define the active set $\mathcal{A}_t = \{i \mid s_i^{(t)} = \text{Active}\}$ and use it to construct forget batches for the next epoch. An ablation study of sample reactivation and an analysis of post-stopping rebound are provided in Appendix E.2.

3.3 Plug-and-Play Unlearning with Active Samples

SWES partitions the active set $\mathcal{A}_t$ into forget batches for epoch $t+1$, so that each active sample participates in forget-side training once during that epoch. Let $\mathcal{B}_f^{(b)} \subseteq \mathcal{A}_t$ denote the b-th nonempty forget batch, with actual batch size $m_b = |\mathcal{B}_f^{(b)}|$. When the base method uses retain data, we randomly sample a retain batch $\mathcal{B}_r^{(b)}$ from $\mathcal{D}_r$ such that $|\mathcal{B}_r^{(b)}| = m_b$. The batch objective preserves the base method's per-sample losses and weights, with both the forget and retain losses averaged over the actual batch size:

$$\mathcal{L}_{\text{batch}}^{(b)}(\theta) = \frac{\lambda_f}{m_b} \sum_{i \in \mathcal{B}_f^{(b)}} \ell_{f,i}^{\text{base}}(\theta) + \frac{\lambda_r}{m_b} \sum_{j \in \mathcal{B}_r^{(b)}} \ell_{r,j}^{\text{base}}(\theta) + \mathcal{R}_{\text{base}}^{(b)}(\theta). \tag{6}$$

Here, $\ell_{f,i}^{\text{base}}$ and $\ell_{r,j}^{\text{base}}$ denote the base method's per-sample forget and retain losses, respectively, and λ_f and λ_r are their original weights. The term $\mathcal{R}_{\text{base}}^{(b)}$ includes any additional regularization or auxiliary losses with their original forms and weights. The retain-loss term is omitted when no retain data are used, and $\mathcal{R}_{\text{base}}^{(b)} = 0$ when no additional losses are present.

By dynamically adjusting sample participation and batch construction, SWES can be integrated with gradient-based unlearning methods that support per-sample loss computation. As the number of active samples decreases, the training workload for both forget samples and the corresponding retain samples also decreases. Stopped samples are excluded from forward and backward passes within training batches, but still undergo the gradient-free state evaluation described in Section 3.2. Therefore, the net computational benefit depends on the training computation saved and the additional monitoring overhead.

4 Experiments

We evaluate SWES on TOFU and MUSE in terms of unlearning effectiveness, model utility, and computational efficiency. Section 4.2 compares the overall performance of the base methods and their SWES variants. Section 4.3 analyzes computational efficiency and compares SWES with global early stopping. Section 4.4 examines the effects of sample removal, stopping criteria, and threshold settings.

4.1 Experimental Setup

Benchmarks. We evaluate SWES on two complementary unlearning benchmarks, TOFU (Maini et al., 2024) and MUSE (Shi et al., 2025), to examine unlearning behavior in controlled and real-

world settings. TOFU consists of synthetic question–answer data about 200 fictitious authors, with the target knowledge introduced into the model through fine-tuning. It provides forget ratios of 1%, 5%, and 10%, offering a controlled setting for analyzing forgetting dynamics. MUSE comprises real-world books and news articles organized into two domains, Books and News. Its forget and retain sets are semantically intertwined, making it a more challenging setting for evaluating unlearning on real-world text.

Baseline Methods. We use Original and Retain as reference models for evaluation. Original is the initial model trained on both the forget and retain sets and serves as the starting point for unlearning. Retain is fine-tuned only on the retain set. We use GradAscent (GA) (Yao et al., 2024) and GradDiff (GD) (Maini et al., 2024) as basic baselines. GA reduces the probability of generating target responses through gradient ascent, while GD additionally incorporates a retention objective to mitigate degradation in model utility. We further integrate SWES into NPO (Zhang et al., 2024), SimNPO (Fan et al., 2025), and WGA (Wang et al., 2025a), and compare each SWES variant with its corresponding base method. Our study focuses on gradient-based unlearning methods. We therefore exclude model editing (Li et al., 2026; Shen et al., 2025; Yu et al., 2025) and in-context intervention methods (Pawelczyk et al., 2024; Wang et al., 2025b; Bhaila et al., 2025) from this comparison. For the main comparisons, we train each baseline and its SWES variant for 10 epochs and report results from the final checkpoint.

Evaluation Metrics. Our metrics cover unlearning effectiveness, model utility, and computational efficiency.

- **TOFU.** We report Forget Quality (FQ), Model Utility (MU), Exact Memorization (EM), Extraction Strength (ES), and Truth Ratio on four evaluation subsets. FQ is the p-value from a Kolmogorov–Smirnov (KS) test comparing the forget-set Truth Ratio distributions of the unlearned model and a reference model trained only on retain data. MU aggregates Probability, ROUGE, and Truth Ratio across non-forget evaluation sets to measure model utility. Following benchmark conventions, higher FQ and MU are better, while lower EM and ES indicate less residual memorization and lower extractability, respectively. F-TR, Ra-TR, R-TR, and Rw-TR correspond to the forget set, real author set, retain set, and real-world knowledge set, respectively.
- **MUSE.** We report knowledge memorization on the retain and forget sets (KM-Dr and KM-Df), verbatim memorization on the forget set (VM-Df), and privacy leakage (PL). Higher KM-Dr indicates better knowledge retention, while lower KM-Df and VM-Df indicate less residual knowledge and verbatim memorization on the forget set, respectively. PL measures privacy leakage based on the deviation of membership inference results from a reference value, with values closer to zero being better.
- **Computational Efficiency.** We report Total BP Saving, Wall Time, and Speedup. Total BP Saving measures the relative reduction in cumulative sample participation in backpropagation across the forget and retain sets, compared with the base method. Wall Time measures the actual training time, including additional overhead such as state monitoring and batch reconstruction. Speedup is the ratio of the base method's training time to that of SWES.

Detailed metric definitions, aggregation procedures, and efficiency measurement protocols are provided in Appendix D.

4.2 Overall Performance

Main Results on TOFU. Table 1 presents the results on TOFU Forget05, from which we draw three observations. First, SWES improved both FQ and MU across all three base methods. Among the compared unlearning methods, SimNPO+SWES achieved the highest FQ of 0.9238, while WGA+SWES achieved the highest MU of 0.6337. These results indicate that reducing the training participation of some samples can improve the trade-off between FQ and MU without compromising either metric in this setting. Second, the effects on residual memorization metrics differed across base methods. With SWES, both EM and ES increased for NPO, whereas EM increased and ES decreased slightly for SimNPO. In contrast, WGA's EM remained largely unchanged, while its ES decreased from 0.0658 to 0.0391, alongside increases in Ra-TR, R-TR, and Rw-TR. Thus, improvements in FQ do not necessarily translate into improvements across all residual memorization

Table 1: Overall performance on TOFU Forget05. + *SWES* denotes the preceding method with SWES. Bold indicates the better result within each pair. Time is wall-clock training time (min).

Method	FQ ↑	MU ↑	EM ↓	ES ↓	F-TR ↑	Ra-TR ↑	R-TR ↑	Rw-TR ↑	Time ↓
Original	0.0013	0.6275	0.9988	0.9808	0.5111	0.6117	0.4570	0.5523	–
Retain	1.0000	0.6268	0.9222	0.7121	0.6777	0.6087	0.4621	0.5612	–
GA	0.0000	0.0000	0.0000	0.0272	0.0000	1.0000	1.0000	1.0000	2.59
GD	0.0000	0.6194	0.0238	0.0272	0.0000	0.7366	0.4721	0.6565	3.52
NPO	0.0680	0.5845	**0.4961**	**0.0517**	0.6600	0.5879	0.4183	**0.5769**	26.80
+ *SWES*	**0.4663**	**0.6090**	0.6957	0.0894	**0.6669**	**0.6373**	**0.4584**	0.5387	**16.49**
SimNPO	0.3935	0.6060	**0.6552**	0.0800	0.6381	0.5924	**0.4443**	0.5760	3.87
+ *SWES*	**0.9238**	**0.6185**	0.7257	**0.0786**	**0.6475**	**0.6227**	0.4319	**0.5946**	**2.12**
WGA	0.1779	0.6244	**0.4525**	0.0658	**0.7015**	0.6062	0.4404	0.5559	13.07
+ *SWES*	**0.8655**	**0.6337**	0.4542	**0.0391**	0.6741	**0.6360**	**0.4516**	**0.5867**	**7.20**

metrics, and the effects of SWES should be assessed across multiple evaluation dimensions. Finally, SWES reduced training time across all three base methods, demonstrating that sample-wise early stopping can deliver practical efficiency gains while improving FQ and MU. Additional results on TOFU Forget01 and Forget10 are provided in Appendix E.1.

Main Results on MUSE. Table 2 summarizes the MUSE results, from which we draw three observations. First, SWES reduced training time across both domains and all three base methods, while improving retain-set performance in most settings. For example, on Books, NPO+SWES increased KM-Dr from 0.6199 to 0.6395 and reduced KM-Df from 0.3787 to 0.3265, although VM-Df increased slightly. These results show that efficiency gains can accompany improvements in knowledge retention and some unlearning metrics. Second, the effects of SWES varied across domains and base methods. On News, NPO+SWES reduced VM-Df while leaving KM-Df largely unchanged; SimNPO+SWES improved KM-Dr but slightly increased both KM-Df and VM-Df. On Books, however, SimNPO+SWES reduced both KM-Df and VM-Df, with a slight decrease in KM-Dr. These results highlight trade-offs between knowledge retention and different dimensions of unlearning performance. Finally, the effects of SWES on WGA differed between the two domains. On Books, KM-Dr increased and PL moved closer to zero, from -28.6058 to -0.4068, while both KM-Df and VM-Df remained at 0.0000. On News, KM-Dr also increased, but VM-Df rose from 0.0161 to 0.2666, and KM-Df increased as well. Thus, SWES delivered efficiency gains across all evaluated settings, although not all unlearning metrics improved simultaneously.

4.3 Computational Efficiency

Training Speedup and Backpropagation Savings. Figure 3 reports the training speedup and Total BP Saving of SWES relative to the corresponding base methods. Across the settings shown in Figure 3—TOFU Forget05, MUSE Books, and MUSE News—SWES reduced cumulative sample participation in backpropagation by 27.5%–64.6% and achieved training speedups of $1.22\times$–$2.34\times$ after accounting for monitoring and other overhead. These results show that dynamically adjusting forget-sample participation and the corresponding number of sampled retain examples can deliver practical efficiency gains across different base methods and data domains. However, a reduction in sample participation does not directly translate into a proportional training speedup. For example, on MUSE Books, NPO+SWES achieved a Total BP Saving of 64.6%, compared with 57.8% for WGA+SWES, yet their speedups were similar at $2.33\times$ and $2.34\times$, respectively. Total BP Saving counts sample participations in backpropagation, whereas wall-clock time also depends on the computational cost of the base method, state monitoring, and data processing. The two metrics therefore provide complementary measures of reduced training participation and practical runtime efficiency.

At an approximately matched backpropagation budget, measured by cumulative sample participation, SWES achieved higher FQ and MU than global early stopping on TOFU Forget05 with NPO as the base method, as detailed in Appendix E.3.

Table 2: Overall performance on MUSE. + *SWES* denotes the preceding method with SWES. Bold indicates the better result within each pair, including ties. Time is wall-clock training time (min).

Method	News					Books				
	KM-Dr↑	KM-Df↓	VM-Df↓	PL→ 0	Time↓	KM-Dr↑	KM-Df↓	VM-Df↓	PL→ 0	Time↓
Original	0.5552	0.6443	0.5789	-99.8111	–	0.6913	0.4712	0.9970	-57.3410	–
Retain	0.5602	0.3279	0.2016	-4.7200	–	0.6874	0.3029	0.1445	8.1600	–
GA	0.0000	0.0000	0.0000	25.1259	87.50	0.0000	0.0000	0.0000	-22.8180	126.60
GD	0.2643	0.3164	0.0073	109.2990	138.08	0.0055	0.0000	0.0000	-24.2141	95.93
NPO	0.4047	**0.4786**	0.3363	-66.6247	108.93	0.6199	0.3787	**0.2208**	-57.8580	154.97
+ *SWES*	**0.4244**	0.4790	**0.3147**	**-65.3023**	**89.34**	**0.6395**	**0.3265**	0.2281	**-56.5089**	**66.55**
SimNPO	0.3621	**0.4752**	**0.3459**	-99.8321	98.01	**0.5610**	0.3086	0.2457	-57.5629	142.15
+ *SWES*	**0.3976**	0.4769	0.3552	**-99.8111**	**75.04**	0.5546	**0.2849**	**0.2288**	**-57.0081**	**82.73**
WGA	0.4001	**0.5014**	**0.0161**	109.8866	99.37	0.2620	**0.0000**	**0.0000**	-28.6058	142.54
+ *SWES*	**0.4350**	0.5098	0.2666	**-92.6952**	**45.61**	**0.2805**	**0.0000**	**0.0000**	**-0.4068**	**60.90**



Figure 3: Training efficiency of SWES relative to the corresponding base methods. TOFU-5%, Books, and News denote TOFU Forget05, MUSE-Books, and MUSE-News, respectively. **Left:** Wall-clock training speedup. **Right:** Total BP Saving (%).

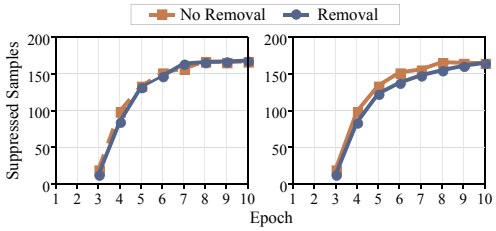


Figure 4: Suppression progress with and without sample removal. **Left:** RLS-based criterion. **Right:** Absolute NLL-based criterion. Curves show the total number of forget samples meeting the corresponding suppression threshold at each epoch.

4.4 Understanding Sample-Wise Stopping

Effect of Sample Removal on Suppression Progress. We examine whether suspending updates after samples reach the suppression threshold affects overall suppression progress. Figure 4 compares continued training (No Removal) with suspending direct updates for samples that meet the threshold (Removal). Under RLS, the two curves closely aligned, with similar numbers of samples meeting the threshold at the end of training. Under Absolute NLL, Removal showed a slight lag midway through training, but the gap gradually narrowed. These results suggest that, in the tested settings, suspending direct updates for samples that have reached the threshold can reduce repeated training while maintaining comparable overall suppression progress.

Comparison of Stopping Criteria. We compare RLS with Absolute NLL, Random, and Second Diff. (Yuan et al., 2025a) on TOFU Forget05 using NPO as the base method. For this comparison, sample reactivation is disabled for all criteria, including RLS. Once a sample is stopped, it remains excluded from backpropagation for the rest of training. We evaluate FQ and MU at different cumulative sample exclusion ratios. Excluded Samples (%) denotes the proportion of forget samples excluded from backpropagation during training. Detailed experimental configurations are provided in Appendix C.1. As shown in the two left panels of Figure 5, RLS achieved higher FQ than Absolute NLL in most settings at comparable sample exclusion ratios, while maintaining similar MU. This suggests that using each sample's initial likelihood as a reference can improve the trade-off between unlearning effectiveness and model utility in the tested settings. The two right panels of

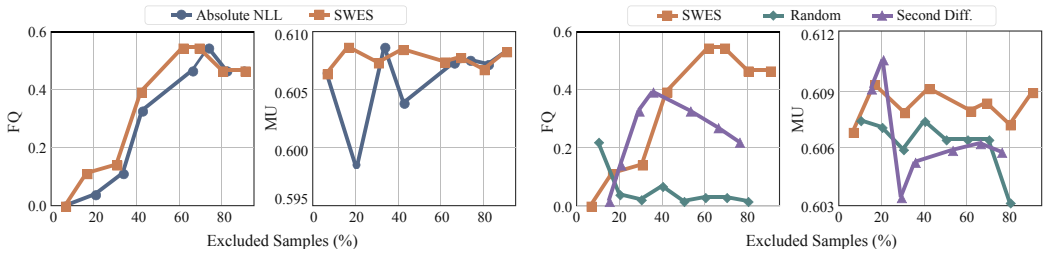


Figure 5: Comparison of sample-wise stopping criteria on TOFU Forget05 with NPO as the base method. Sample reactivation is disabled for all criteria. **Left:** Comparison of RLS and Absolute NLL. **Right:** Comparison of RLS, Random, and Second Diff. Each pair shows FQ and MU as functions of the cumulative sample exclusion ratio.

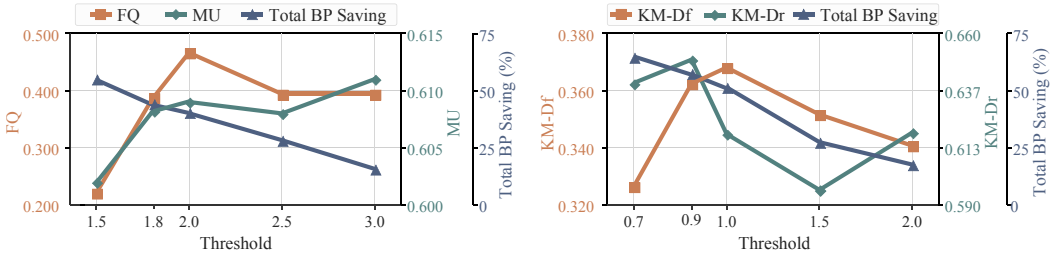


Figure 6: Sensitivity of SWES to the stopping threshold τ_{stop} with NPO as the base method. **Left:** FQ, MU, and Total BP Saving on TOFU Forget05. **Right:** KM-Df, KM-Dr, and Total BP Saving on MUSE Books.

Figure 5 show that Random yielded lower FQ in most settings, indicating that random sample exclusion alone is insufficient to achieve comparable unlearning effectiveness. As the exclusion ratio increased, FQ under Second Diff. initially rose and then gradually declined. In contrast, RLS maintained relatively high FQ at moderate and high exclusion ratios while preserving comparable MU, supporting its effectiveness as a sample-wise stopping criterion.

Sensitivity to the Stopping Threshold. To examine the sensitivity of SWES to the stopping threshold, we use NPO as the base method and evaluate unlearning effectiveness, model utility, and Total BP Saving. We consider $\tau_{\text{stop}} \in \{1.5, 1.8, 2.0, 2.5, 3.0\}$ on TOFU Forget05 and $\tau_{\text{stop}} \in \{0.7, 0.9, 1.0, 1.5, 2.0\}$ on MUSE Books. As shown in Figure 6, Total BP Saving gradually decreased on both benchmarks as the threshold increased, indicating that stricter stopping conditions lead to fewer sample exclusions. On TOFU Forget05, FQ peaked at $\tau_{\text{stop}} = 2.0$, while MU varied only slightly overall. On MUSE Books, KM-Df was lowest at $\tau_{\text{stop}} = 0.7$, whereas KM-Dr was highest at $\tau_{\text{stop}} = 0.9$. Compared with a threshold of 0.7, a threshold of 0.9 provided better knowledge retention but resulted in higher KM-Df and lower computational savings. These results illustrate the trade-offs among unlearning effectiveness, knowledge retention, and computational efficiency.

5 Conclusion

Differences in forgetting progress across samples can lead to redundant updates under a fixed training budget. We propose Sample-Wise Early Stopping (SWES), which adaptively controls sample participation in LLM unlearning. SWES combines RLS with temporal smoothing and threshold-based control to suspend or resume backpropagation for individual samples while preserving the base method's per-sample losses and objective weights. Experiments on TOFU and MUSE show computational savings and practical training speedups after accounting for overhead, alongside improved trade-offs between unlearning effectiveness and model utility in multiple settings.

AI USE STATEMENT

We used generative AI tools to assist with translation and language editing, provide feedback on research methodology and experimental design, support code-related tasks, and assist in interpreting experimental results. All AI-assisted outputs, implementations, calculations, citations, and claims were independently reviewed and verified by the authors, who take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

The SWES criterion, sample-wise controller, and batch construction are described in Section 3. Appendix C.1 reports the experimental environment, training settings, hyperparameter search spaces, and configurations for the sample-removal and stopping-criterion experiments. Detailed metric definitions and efficiency measurement protocols are provided in Appendix D. We provide our implementation as supplementary material.

ETHICS STATEMENT

This work studies efficient LLM unlearning, with potential applications in privacy protection and responsible data management. Our empirical results do not establish guarantees of complete information removal.

REFERENCES

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, 2003.

Karuna Bhaila, Minh-Hao Van, and Xintao Wu. Soft prompting for unlearning in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pp. 4046–4056. Association for Computational Linguistics, 2025.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 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.

Rich Caruana, Steve Lawrence, and C. Lee Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pp. 402–408. MIT Press, 2000.

Sungmin Cha, Sungjun Cho, Dasol Hwang, and Moontae Lee. Towards robust and parameter-efficient knowledge unlearning for llms. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zico Kolter, Zachary C. Lipton, and Pratyush Maini. Openunlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

Guangyao Dou, Zheyuan Liu, Qing Lyu, Kaize Ding, and Eric Wong. Avoiding copyright infringement via large language model unlearning. In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, volume NAACL 2025 of *Findings of ACL*, pp. 5176–5200. Association for Computational Linguistics, 2025.

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

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

Xiaohua Feng, Yuyuan Li, Chengye Wang, Junlin Liu, Li Zhang, and Chaochao Chen. A neuro-inspired interpretation of unlearning in large language models through sample-level unlearning difficulty. *arXiv preprint arXiv:2504.06658*, 2025.

Federico Di Gennaro, Alexander Shevchenko, and Fanny Yang. Fast unlearning at scale via margin self-correction. *arXiv preprint arXiv:2606.02920*, 2026.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 7403–7412. Association for Computational Linguistics, 2023.

Aravind Krishnan, Siva Reddy, and Marius Mosbach. Not all data are unlearned equally. *arXiv preprint arXiv:2504.05058*, 2025.

Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pp. 3637–3647. ACM, 2023.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Pre-trained language models for text generation: A survey. *ACM Comput. Surv.*, 56(9):230:1–230:39, 2024.

Zexi Li, Xiangzhu Wang, William F. Shen, Meghdad Kurmanji, Xinchi Qiu, Dongqi Cai, Chao Wu, and Nicholas D. Lane. Editing as unlearning: Are knowledge editing methods strong baselines for large language model unlearning? In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 37627–37635. AAAI Press, 2026.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. Rethinking machine unlearning for large language models. *Nat. Mac. Intell.*, 7(2): 181–194, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

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

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David A. Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35181–35224. PMLR / OpenReview.net, 2024.

Anmol Reddy Mekala, Vineeth Dorna, Shreya Dubey, Abhishek Lalwani, David Koleczek, Mukund Rungta, Sadid A. Hasan, and Elita A. Lobo. Alternate preference optimization for unlearning factual knowledge in large language models. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pp. 3732–3752. Association for Computational Linguistics, 2025.

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Florian Tramèr, and Katherine Lee. Scalable

extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few-shot unlearners. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 40034–40050. PMLR / OpenReview.net, 2024.

Lutz Prechelt. Early stopping - but when? In *Neural Networks: Tricks of the Trade - Second Edition*, volume 7700 of *Lecture Notes in Computer Science*, pp. 53–67. Springer, 2012.

Pengyang Shao, Naixin Zhai, Lei Chen, Yonghui Yang, Fengbin Zhu, Xun Yang, and Meng Wang. Baldro: A distributionally robust optimization based framework for large language model unlearning. In *Proceedings of the ACM Web Conference 2026, WWW 2026, Dubai, United Arab Emirates, originally scheduled for April 13-17, 2026, rescheduled for June 29 - July 3, 2026*, pp. 8874–8884. ACM, 2026.

William F. Shen, Xinchi Qiu, Meghdad Kurmanji, Alex Iacob, Lorenzo Sani, Yihong Chen, Nicola Cancedda, and Nicholas D. Lane. LLM unlearning via neural activation redirection. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

Shaojie Shi, Xiaoyu Tan, Xihe Qiu, Chao Qu, Kexin Nie, Yuan Cheng, Wei Chu, Yinghui Xu, and Yuan Qi. ULMR: unlearning large language models via negative response and model parameter average. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP 2024 - Industry Track, Miami, Florida, USA, November 12-16, 2024*, pp. 755–762. Association for Computational Linguistics, 2024.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. MUSE: machine unlearning six-way evaluation for language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Dean L. Slack and Noura Al Moubayed. Early detection and reduction of memorization for domain adaptation and instruction tuning. *Trans. Assoc. Comput. Linguistics*, 13:1459–1473, 2025.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 14918–14937. Association for Computational Linguistics, 2023.

Bozhong Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Mengru Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. To forget or not? towards practical knowledge unlearning for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pp. 1524–1537. Association for Computational Linguistics, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q. Weinberger. Rethinking LLM unlearning objectives: A gradient perspective and go beyond. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a.

Yaxuan Wang, Chris Yuhao Liu, Quan Liu, Jinglong Pang, Wei Wei, Yujia Bao, and Yang Liu. DRAGON: guard LLM unlearning in context via negative detection and reasoning. *arXiv preprint arXiv.2511.05784*, 2025b.

Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Shah, Yujia Bao, Yang Liu, and Wei Wei. LLM unlearning via loss adjustment with only forget data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025c.

Yue Wang, Qizhou Wang, Feng Liu, Wei Huang, Yali Du, Xiaojiang Du, and Bo Han. GRU: mitigating the trade-off between unlearning and retention for llms. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025d.

Puning Yang, Qizhou Wang, Zhuo Huang, Tongliang Liu, Chengqi Zhang, and Bo Han. Exploring criteria of loss reweighting to enhance LLM unlearning. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Miao Yu, Liang Lin, Guibin Zhang, Xinfeng Li, Junfeng Fang, Ningyu Zhang, Kun Wang, and Yang Wang. Unierase: Unlearning token as a universal erasure primitive for language models. *arXiv preprint arXiv.2505.15674*, 2025.

Suqin Yuan, Lei Feng, and Tongliang Liu. Early stopping against label noise without validation data. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Suqin Yuan, Runqi Lin, Lei Feng, Bo Han, and Tongliang Liu. Instance-dependent early stopping. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a.

Suqin Yuan, Muyang Li, Lei Feng, Bo Han, Gang Niu, Masashi Sugiyama, and Tongliang Liu. Early stopping without validation data in weakly supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–16, 2026a.

Suqin Yuan, Runqi Lin, Felix Azian, Lei Feng, Bo Han, Gang Niu, Masashi Sugiyama, and Tongliang Liu. Instance-dependent early stopping for adaptive data pruning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(10):11577–11592, 2026b.

Xiaojian Yuan, Tianyu Pang, Chao Du, Kejiang Chen, Weiming Zhang, and Min Lin. A closer look at machine unlearning for large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b.

ChengXiang Zhai. Large language models and future of information retrieval: Opportunities and challenges. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pp. 481–490. ACM, 2024.

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

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Appendix of
Sample-Wise Early Stopping for LLM Unlearning

CONTENTS

A Notations **15**

B Additional Method Details **15**

B.1 Gradient-Based Unlearning Methods 15

B.2 Pseudocode of SWES 16

C Detailed Experimental Setup **16**

C.1 Experimental Environment and Implementation 16

D Evaluation Metrics and Efficiency Measurement **18**

D.1 Evaluation Metrics on TOFU 18

D.2 Evaluation Metrics on MUSE 18

D.3 Computational Efficiency Measurement 19

E Supplementary Results and Analysis **19**

E.1 Additional Results on TOFU 19

E.2 Effect of Sample Reactivation 19

E.3 Comparison with Global Early Stopping 20

F Limitations **22**

A NOTATIONS

In this section, we summarize the main notation in Table 3.

Table 3: List of main symbols used throughout the paper.

Notation	Description
$\mathcal{D}_f,\ \mathcal{D}_r$	Forget and retain datasets
N_f	Number of forget samples
(x_i, y_i)	Input–response pair of sample i
$y_{i,j},\ y_{i,<j}$	The j-th response token and its preceding prefix
$\lvert y_i \rvert$	Response length
θ_0	Model parameters before unlearning
θ_t	Model parameters after epoch t
$p_\theta(y_i \mid x_i)$	Response likelihood
$\ell_i(\theta)$	Length-normalized negative log-likelihood
$\rho_i^{(t)}$	Reference-relative likelihood suppression (RLS)
$\bar{\rho}_i^{(t)}$	Smoothed RLS
T	Fixed training budget in epochs
W	Smoothing window size
τ_{stop}	Stopping threshold
$s_i^{(t)}$	Sample state: Active or Stopped
$\mathcal{A}_t$	Active sample indices for epoch $t+1$
$\mathcal{S}_t$	Stopped sample indices for epoch $t+1$
$\mathcal{B}_f^{(b)},\ \mathcal{B}_r^{(b)}$	Forget batch and matched retain batch
m_b	Actual batch size
$\ell_{f,i}^{\text{base}},\ \ell_{r,j}^{\text{base}}$	Per-sample forget and retain losses
$\mathcal{L}_{\text{batch}}^{(b)}$	Batch training objective
$\lambda_f,\ \lambda_r$	Forget and retain loss coefficients
$\mathcal{R}_{\text{base}}^{(b)}$	Regularization and auxiliary losses
$\mathcal{L}_f,\ \mathcal{L}_r$	Forget and retain objectives
λ	Relative retain-loss weight
$\beta_{\text{NPO}},\ \beta_{\text{Sim}}$	NPO and SimNPO logistic parameters
α	WGA weighting exponent

B ADDITIONAL METHOD DETAILS

B.1 GRADIENT-BASED UNLEARNING METHODS

We review several representative gradient-based LLM unlearning methods considered in this work. A common formulation combines losses on the forget set $\mathcal{D}_f$ and the retain set $\mathcal{D}_r$ (Maini et al., 2024; Fan et al., 2025):

$$\mathcal{L}_{\text{all}}(\theta) = \mathcal{L}_f(\theta) + \lambda \mathcal{L}_r(\theta), \tag{7}$$

where $\mathcal{L}_f$ and $\mathcal{L}_r$ denote the forget and retain losses, respectively, and $\lambda \geq 0$ controls their relative contributions. The model parameters θ are updated by minimizing this objective.

Gradient ascent (GA) (Yao et al., 2024) is a classical approach that maximizes the negative log-likelihood of forget responses. Under the minimization formulation above, its equivalent forget objective is

$$\mathcal{L}_f^{\text{GA}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_f}\left[\log p_\theta(y \mid x)\right]. \tag{8}$$

Combining this forget objective with a negative log-likelihood loss on retain data yields gradient difference (GD) (Maini et al., 2024). Although GA suppresses target likelihoods, continued updates can also impair model utility (Zhang et al., 2024; Wang et al., 2025a). The retain loss mitigates this degradation but does not explicitly determine when individual forget samples should stop participating in training.

To regulate forget updates, negative preference optimization (NPO) (Zhang et al., 2024) uses a fixed reference model p_{θ_0}, where θ_0 denotes the model parameters before unlearning:

$$\mathcal{L}_f^{\mathrm{NPO}}(\theta) = -\frac{2}{\beta_{\mathrm{NPO}}}\mathbb{E}_{(x,y)\sim\mathcal{D}_f}\left[\log\sigma\left(-\beta_{\mathrm{NPO}}\log\frac{p_\theta(y\mid x)}{p_{\theta_0}(y\mid x)}\right)\right], \tag{9}$$

where $\sigma(z) = (1+e^{-z})^{-1}$ and $\beta_{\mathrm{NPO}} > 0$ controls the sharpness of the logistic transformation. NPO induces sample-dependent gradient weights based on the current-to-reference likelihood ratio. As this ratio decreases, the update weight for the corresponding sample also decreases.

SimNPO (Fan et al., 2025) removes this dependence on a reference model and instead uses length-normalized log-likelihoods. Its zero-margin forget objective is

$$\mathcal{L}_f^{\mathrm{SimNPO}}(\theta) = -\frac{2}{\beta_{\mathrm{Sim}}}\mathbb{E}_{(x,y)\sim\mathcal{D}_f}\left[\log\sigma\left(-\frac{\beta_{\mathrm{Sim}}}{|y|}\log p_\theta(y\mid x)\right)\right], \tag{10}$$

where $|y|$ denotes the response length and $\beta_{\mathrm{Sim}} > 0$ controls the logistic transformation. The resulting weights depend on the current response likelihood and length, without requiring reference-model likelihoods for the forget objective.

Weighted gradient ascent (WGA) (Wang et al., 2025a) regulates updates at the token level. It uses confidence-dependent weights to counteract the inverse-likelihood factor in the GA gradient. Specifically, the weight assigned to the k-th response token is

$$w_{x,y,k}^{\mathrm{WGA}} = \mathrm{sg}\left[p_\theta(y_k\mid x, y_{<k})^\alpha\right],$$

where $\alpha > 0$ controls the weighting strength and $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator. The weights are updated using the current predictions but treated as constants during backpropagation. The forget objective is

$$\mathcal{L}_f^{\mathrm{WGA}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_f}\left[\sum_{k=1}^{|y|} w_{x,y,k}^{\mathrm{WGA}}\log p_\theta(y_k\mid x, y_{<k})\right]. \tag{11}$$

This token-level weighting reduces the contributions of target tokens that already have low prediction probabilities.

B.2 Pseudocode of SWES

Algorithm 1 summarizes the SWES framework. The function $\ell_i(\theta)$ denotes the length-normalized negative log-likelihood defined in the main text.

C Detailed Experimental Setup

C.1 Experimental Environment and Implementation

Experimental Environment. Unless otherwise specified, all experiments are conducted on a server equipped with a single NVIDIA A100-SXM4 GPU (80 GB of memory). Our implementation uses Python 3.12, PyTorch 2.10.0, and Transformers 4.51.3 with CUDA 12.6 on Ubuntu 22.04.

Training Configuration. Our experimental setup follows OpenUnlearning (Dorna et al., 2025) and Wang et al. (2025d), with the configurations detailed below. We use the AdamW optimizer (Loshchilov & Hutter, 2019) across all benchmarks and LLaMA2-7B (Touvron et al., 2023) as the primary backbone. For the main comparisons on TOFU and MUSE, we train each baseline and its SWES variant for 10 epochs and report all results from the final checkpoint. Within each baseline–SWES pair, both methods use the same batch size. We perform a grid search over learning rates in $\{1\times10^{-5}, 2\times10^{-5}, 5\times10^{-5}, 1\times10^{-4}\}$ and batch sizes in $\{8, 16, 32\}$. For SWES, we additionally search over stopping thresholds $\tau_{\mathrm{stop}} \in \{1.5, 1.8, 2.0, 2.5, 3.0\}$ on TOFU and $\tau_{\mathrm{stop}} \in \{0.7, 0.9, 1.0, 1.5, 2.0\}$ on MUSE. For both benchmarks, we select configurations from the evaluated grid by jointly considering unlearning effectiveness and model utility at the final checkpoint. The temporal smoothing window is fixed at $W = 3$ in all SWES experiments. Unless otherwise specified, all remaining training settings, including the learning-rate scheduler and

Algorithm 1 Sample-Wise Early Stopping (SWES)

Require: Forget set $\mathcal{D}_f$, optional retain set $\mathcal{D}_r$, initial parameters θ_0
Require: Base method M, batch size B, smoothing window W, threshold $\tau_{\text{stop}} > 0$, maximum epochs T
Ensure: Unlearned model parameters θ

$\mathcal{I} \leftarrow \{1, \ldots, N_f\}$, $\theta \leftarrow \theta_0$
$s_i^{(0)} \leftarrow$ Active for all $i \in \mathcal{I}$
Compute and cache reference NLL $b_i \leftarrow \ell_i(\theta_0)$ for all $i \in \mathcal{I}$ without gradients
for $t = 1, \ldots, T$ **do**
    $\mathcal{A}_{t-1} \leftarrow \{i \in \mathcal{I} \mid s_i^{(t-1)} = \text{Active}\}$
    Shuffle and partition $\mathcal{A}_{t-1}$ into forget batches of size at most B
    for all forget batches $\mathcal{B}_f$ **do**
        Compute the mean per-sample forget loss of M over $\mathcal{B}_f$
        if M uses retain data **then**
            Randomly sample $\mathcal{B}_r$ from $\mathcal{D}_r$ such that $|\mathcal{B}_r| = |\mathcal{B}_f|$
            Compute the mean per-sample retain loss of M over $\mathcal{B}_r$
        end if
        Form $\mathcal{L}_M$ using the original objective terms and weights of M
        Update θ by one optimization step of M using $\mathcal{L}_M$
    end for
    $\theta_t \leftarrow \theta$ ▷ Unchanged if $\mathcal{A}_{t-1} = \emptyset$
    Evaluate all samples, including stopped samples, without gradients: $\rho_i^{(t)} \leftarrow \ell_i(\theta_t) - b_i$, $i \in \mathcal{I}$
    for all $i \in \mathcal{I}$ **do**
        if $t \geq W$ **then**
            $\bar{\rho}_i^{(t)} \leftarrow \dfrac{1}{W} \displaystyle\sum_{k=t-W+1}^{t} \rho_i^{(k)}$
            if $\bar{\rho}_i^{(t)} \geq \tau_{\text{stop}}$ **then**
                $s_i^{(t)} \leftarrow$ Stopped
            else
                $s_i^{(t)} \leftarrow$ Active
            end if
        else
            $s_i^{(t)} \leftarrow$ Active
        end if
    end for
end for
return θ

method-specific hyperparameters, follow the defaults in the official OpenUnlearning implementation.

Sample-Removal Experiment. For the sample-removal experiment in Figure 4, we compare RLS and Absolute NLL on TOFU Forget05, evaluating each criterion with and without sample removal. Both criteria use a threshold of 2.0. All runs use a learning rate of 2×10^{-5}, a batch size of 8, and a training budget of 10 epochs. At each epoch, we record the number of forget samples that satisfy the corresponding threshold.

Stopping-Criterion Comparisons. We compare RLS, Absolute NLL, Random, and Second Diff. on TOFU Forget05 using NPO as the base method.

Random. At initialization, we generate a single random permutation of all forget sample indices. A predefined non-decreasing sequence $\{K_t\}_{t=1}^{T}$ specifies the cumulative number of stopped samples at the end of each epoch. After epoch t, the first K_t samples in this permutation are excluded from subsequent training. The stopped set therefore grows monotonically, and stopped samples are not reactivated. Sample selection is independent of per-sample loss values and suppression scores.

Second Diff. We adapt the second-order loss-difference criterion from IES (Yuan et al., 2025a) to the length-normalized NLL of each forget sample. Let $\ell_i^{(t)} = \ell_i(\theta_t)$. We compute

$$d_i^{(t)} = \ell_i^{(t)} - 2\ell_i^{(t-1)} + \ell_i^{(t-2)}, \qquad c_i^{(t)} = \frac{1}{W} \sum_{j=0}^{W-1} \left| d_i^{(t-j)} \right|, \tag{12}$$

where $W = 3$. Once $c_i^{(t)} < \delta$, sample i is excluded from all subsequent training epochs and is not reactivated.

D Evaluation Metrics and Efficiency Measurement

D.1 Evaluation Metrics on TOFU

We follow the TOFU evaluation protocol (Maini et al., 2024) and use the metric implementations provided by OpenUnlearning (Dorna et al., 2025).

Forget Quality (FQ). FQ compares the unlearned model with a reference model trained only on retain data. It is the p-value of a two-sample Kolmogorov–Smirnov test comparing their Truth Ratio distributions on the same forget set. Following the benchmark convention, higher FQ is preferred.

Model Utility (MU). MU summarizes model performance on three non-forget subsets: the retain set, real author set, and real-world knowledge set. Each subset is evaluated using Probability, ROUGE-L recall, and Truth Ratio. The resulting nine scores are aggregated using the harmonic mean, with higher MU indicating better preservation of model utility.

Exact Memorization (EM). EM measures the fraction of correctly predicted target tokens conditioned on the ground-truth preceding context. Token accuracy is computed per sample and then averaged across the evaluated subset. Lower EM on the forget set indicates less residual token-level memorization.

Extraction Strength (ES). ES measures how much of a target answer can be recovered after supplying a sufficient prefix. A shorter prefix required for exact reconstruction corresponds to a higher sample-level score. Scores are averaged across the evaluated subset, with lower ES on the forget set indicating lower extractability.

Truth Ratio (TR). TR compares the length-normalized likelihoods of correct answers and perturbed incorrect alternatives. We report F-TR, Ra-TR, R-TR, and Rw-TR on the forget, real author, retain, and real-world knowledge sets, respectively.

For detailed metric definitions and implementation details, we refer readers to TOFU (Maini et al., 2024) and OpenUnlearning (Dorna et al., 2025).

D.2 Evaluation Metrics on MUSE

We follow the MUSE evaluation protocol (Shi et al., 2025) and use the metric implementations provided by OpenUnlearning (Dorna et al., 2025).

Knowledge Memorization (KM). KM evaluates the model's ability to answer questions about the evaluated corpus, measured by ROUGE-L F1 between generated and reference answers. We report KM-Dr and KM-Df on the retain and forget sets, respectively. Higher KM-Dr indicates better knowledge retention, while lower KM-Df indicates less residual knowledge of the forget set.

Verbatim Memorization (VM). VM measures the model's tendency to reproduce forget-set text when prompted with its prefix. We compute ROUGE-L F1 between generated and reference continuations and report the average score as VM-Df. Lower values indicate less verbatim memorization.

Privacy Leakage (PL). PL evaluates whether forget-set samples remain distinguishable from held-out samples through membership inference. It measures the relative deviation of the attack AUC from a reference AUC under the MUSE evaluation protocol. Values closer to zero indicate closer agreement with the reference privacy behavior, while large deviations in either direction can indicate privacy leakage.

For detailed metric definitions, we refer readers to MUSE (Shi et al., 2025).

D.3 Computational Efficiency Measurement

We evaluate the computational efficiency of SWES relative to each corresponding base method using Total BP Saving, Wall Time, and Speedup.

Total BP Saving. Total BP Saving measures the relative reduction in the combined number of sample participations in backpropagation across the forget and retain sets. For method m, let $B_{m,f}$ and $B_{m,r}$ denote the cumulative BP participation counts for forget and retain samples, respectively. We define

$$\text{Total BP Saving} = \left(1 - \frac{B_{\text{SWES},f} + B_{\text{SWES},r}}{B_{\text{base},f} + B_{\text{base},r}}\right) \times 100\%. \tag{13}$$

For methods without retain-set training, $B_{m,r} = 0$. Higher values indicate greater reductions in sample-level backward-pass workload.

Each occurrence of a sample in backpropagation is counted separately, including repeated sampling within or across microbatches and epochs. When gradient accumulation is used, samples are counted in each contributing microbatch. Suspended samples contribute no BP count while inactive; their subsequent training occurrences are counted if they are reactivated. Retain samples are counted according to their actual participation in backpropagation, including size-matched retain sampling when applicable. Gradient-free monitoring and reference-model forward passes are excluded from the BP count.

Wall Time. Wall Time measures the elapsed training time, including parameter updates and the additional overhead introduced by SWES. This overhead includes the one-time precomputation of reference NLLs, gradient-free evaluation of the forget set, sample-state monitoring, and batch reconstruction. Reference NLL precomputation and gradient-free monitoring contribute to Wall Time but are excluded from the BP count.

Speedup. We define training speedup as

$$\text{Speedup} = \frac{T_{\text{base}}}{T_{\text{SWES}}}, \tag{14}$$

where T_{base} and T_{SWES} denote the Wall Time of the base method and its SWES variant, respectively. Both are measured using the same timing boundaries and hardware configuration. A speedup greater than 1 indicates faster training.

E Supplementary Results and Analysis

This section presents supplementary experimental results and analyses that complement the main findings.

E.1 Additional Results on TOFU

We compared the baseline methods with their SWES variants on TOFU Forget01 and Forget10, corresponding to forget-set ratios of 1% and 10%, respectively, using the same backbone model and evaluation metrics as in the main experiments. As shown in Tables 4 and 5, SWES reduced training time across all six comparisons, with speedups of 1.15–$2.01\times$ and Total BP Saving ranging from 15.9% to 59.8%. It also consistently reduced EM and ES while improving or maintaining FQ. MU improved in five of the six comparisons, with only a slight decrease for WGA on Forget01 (from 0.6221 to 0.6219). However, FQ remained low for NPO and WGA on Forget10, and changes in the subset-specific Truth Ratio scores were mixed.

E.2 Effect of Sample Reactivation

We evaluated the contribution of sample reactivation by comparing SWES with No Reactivation, a variant that permanently excludes samples from backpropagation once they satisfy the stopping criterion. We used NPO on TOFU Forget05 with $W = 3$ and $\tau_{\text{stop}} \in \{1.5, 1.8, 2.0, 2.5\}$, with one run per configuration. As shown in Table 6, reactivation improved FQ at all four thresholds, with absolute gains of 0.0728–0.1730. MU improved at three thresholds, with a slight decrease of 0.0009

Table 4: Overall performance on TOFU Forget01. + *SWES* denotes the preceding method with SWES. Bold indicates the better result within each pair, including ties. Time is wall-clock training time (min). BP denotes total BP saving (%).

Method	FQ ↑	MU ↑	EM ↓	ES ↓	F-TR ↑	Ra-TR ↑	R-TR ↑	Rw-TR ↑	Time ↓	BP ↑
Original	0.0030	0.6275	1.0000	1.0000	0.5317	0.6117	0.4596	0.5523	–	–
Retain	1.0000	0.6268	0.7120	0.0719	0.6777	0.6087	0.4621	0.5612	–	–
GA	0.0286	0.5676	0.6770	0.0787	0.5177	0.6134	0.4632	0.5751	1.78	–
GD	0.0286	0.5933	0.6883	0.0902	0.5084	0.6186	0.4620	0.5703	1.13	–
NPO	**0.5786**	0.6171	0.7473	0.0785	**0.6731**	0.6004	0.4457	0.5639	5.40	–
+ *SWES*	**0.5786**	**0.6287**	**0.5579**	**0.0454**	0.6379	**0.6234**	**0.4495**	**0.5756**	**2.94**	45.3
SimNPO	0.1650	0.5856	0.6536	0.0474	**0.6792**	0.5860	**0.4391**	0.5565	1.03	–
+ *SWES*	**0.7659**	**0.6152**	**0.3965**	**0.0371**	0.5717	**0.6960**	0.4071	**0.6473**	**0.79**	31.6
WGA	0.1650	**0.6221**	0.6279	0.0585	**0.6478**	0.6011	**0.4543**	0.5506	5.17	–
+ *SWES*	**0.5786**	0.6219	**0.4429**	**0.0415**	0.5600	**0.6403**	0.4500	**0.5790**	**2.57**	59.8

Table 5: Overall performance on TOFU Forget10. + *SWES* denotes the preceding method with SWES. Bold indicates the better result within each pair. Time denotes wall-clock training time (min). BP denotes total BP saving (%).

Method	FQ↑	MU↑	EM↓	ES↓	F-TR↑	Ra-TR↑	R-TR↑	Rw-TR↑	Time↓	BP↑
Original	1.98e−25	0.6275	0.9988	0.9820	0.5181	0.6117	0.4596	0.5523	–	–
Retain	1.0000	0.6130	0.6653	0.0696	0.6808	0.5848	0.4572	0.5329	–	–
GA	1.06e−239	0.0000	0.0141	0.0269	4.06e−27	1.0000	1.0000	1.0000	6.72	–
GD	3.15e−15	0.5441	0.7186	0.1043	0.5114	0.5653	0.4858	0.5313	7.74	–
NPO	9.53e−5	0.6320	0.6955	0.0824	0.6083	**0.6522**	0.4674	**0.5699**	52.74	–
+ *SWES*	**0.0030**	**0.6324**	**0.6859**	**0.0726**	**0.6125**	0.6482	**0.4677**	0.5622	**29.43**	47.0
SimNPO	0.0446	0.6279	0.6494	0.0688	0.6235	0.6567	**0.4421**	0.5907	8.44	–
+ *SWES*	**0.2107**	**0.6291**	**0.5472**	**0.0600**	**0.6329**	**0.6657**	0.4365	**0.6115**	**5.40**	42.4
WGA	3.20e−18	0.6216	0.8579	0.2022	0.5662	0.6100	0.4481	0.5495	27.49	–
+ *SWES*	**4.64e−12**	**0.6228**	**0.7427**	**0.1118**	**0.5823**	**0.6220**	**0.4530**	**0.5603**	**23.82**	15.9

at $\tau_{\text{stop}} = 1.8$. The effects on EM and ES were mixed. These gains required additional sample participation in backpropagation: compared with No Reactivation, Total BP Saving decreased by 0.85–1.80 percentage points, while SWES still achieved savings of 28.30%–55.10%.

Figure 7 examines post-stopping rebound at $\tau_{\text{stop}} = 2.0$. Here, rebound is defined as a sample's smoothed RLS falling below the stopping threshold after the sample has stopped. We tracked the same six samples that stopped in both runs and exhibited rebound in at least one run. With SWES, the number of tracked samples with smoothed RLS below the threshold decreased from five at epoch 7 to zero at epoch 10; without reactivation, all six were below the threshold at the end of training. The final mean smoothed RLS was 2.56 with SWES and 1.43 without reactivation. These trajectories illustrate that reaching the stopping threshold once does not guarantee sustained suppression and that reactivation can help restore suppression after rebound for the tracked samples.

E.3 Comparison with Global Early Stopping

To examine the role of sample-wise early stopping, we compared SWES with global early stopping after epochs 5, 6, and 7, using NPO on TOFU Forget05. Stopping after epoch 6 approximately matched SWES's cumulative budget for sample participation in backpropagation. As shown in

Table 6: Ablation of sample reactivation on TOFU Forget05 using NPO as the base method. Bold indicates the best result within each threshold pair. Total BP Saving accounts for cumulative sample participation in backpropagation on both forget and retain data.

τ_{stop}	Method	FQ ↑	MU ↑	EM ↓	ES ↓	Total BP Saving (%) ↑
1.5	No Reactivation	0.1421	0.5999	0.7256	0.1077	**56.15**
	SWES	**0.2205**	**0.6017**	**0.7251**	**0.1038**	55.10
1.8	No Reactivation	0.2205	**0.6065**	0.6974	0.0980	**47.90**
	SWES	**0.3935**	0.6056	**0.6951**	**0.0910**	46.10
2.0	No Reactivation	0.3935	0.6082	0.6958	**0.0874**	**41.95**
	SWES	**0.4663**	**0.6090**	**0.6957**	0.0894	41.10
2.5	No Reactivation	0.2705	0.6034	**0.6968**	**0.0842**	**29.40**
	SWES	**0.3935**	**0.6080**	0.6975	0.0846	28.30

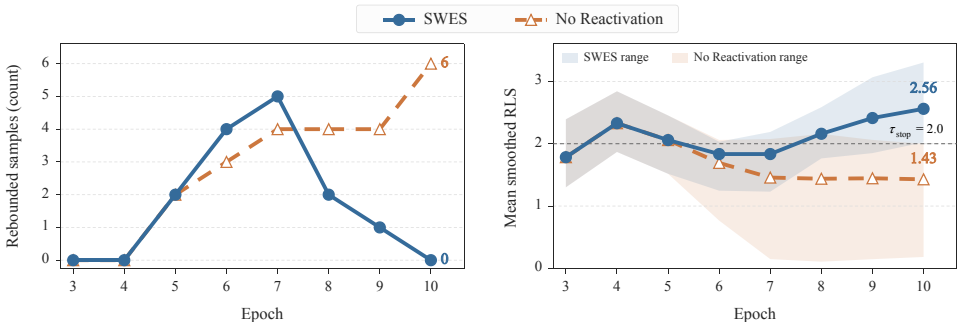


Figure 7: Sample reactivation on TOFU Forget05 with NPO ($\tau_{\text{stop}} = 2.0$, $W = 3$). Both panels track six samples that stopped in both runs and rebounded in at least one. **Left:** Current count below threshold, including reactivated samples. **Right:** Mean smoothed RLS with min–max shading. The horizontal dashed line marks the stopping threshold.

Table 7, GS @ E6 and NPO+SWES both performed 150 training steps, with Total BP Saving values of 40.0% and 41.1%, respectively. Despite using slightly fewer sample participations in backpropagation, NPO+SWES achieved higher FQ (0.4663 vs. 0.1123) and MU (0.6090 vs. 0.5712) than GS @ E6. These results indicate that, under approximately matched budgets in this setting, dynamically allocating training participation based on sample suppression states yields better forgetting quality and model utility than stopping all samples at a common epoch.

Table 7: Global early stopping (GS) versus SWES on TOFU Forget05 with NPO. GS @ E6 approximately matches SWES's cumulative sample participation in backpropagation.

Method	Steps	Total BP Saving (%)	MU (↑)	FQ (↑)
NPO	250	0.0	0.5845	0.0680
GS @ E5	125	50.0	0.5611	0.0680
GS @ E6	150	40.0	0.5712	0.1123
GS @ E7	175	30.0	0.5793	0.0085
NPO + SWES	150	41.1	**0.6090**	**0.4663**

F LIMITATIONS

SWES exhibits sensitivity to the suppression threshold, which may require tuning across datasets and unlearning methods. Future work will explore more effective threshold selection strategies to better balance unlearning effectiveness and model utility. In addition, although experiments support the effectiveness of RLS-based stopping, formal guarantees for its forgetting–utility trade-off remain to be established.