SAUL: Sharpness-Aware Augmented-Lagrangian Unlearning

Jaewan Choi[1*], **Junyoung Yang**[1*], **Sangdon Park**[1,2]
[1]Computer Science and Engineering, POSTECH
[2]Graduate School of Artificial Intelligence, POSTECH
{cjw7656, sheepjun0330, sangdon}@postech.ac.kr
[*]Equal contribution.

Abstract

Machine unlearning in Large Language Models (LLMs) faces a critical trade-off between erasing target knowledge and preserving general utility. We propose **SAUL** (Sharpness-Aware Augmented-Lagrangian Unlearning), which formulates unlearning as a constrained minimization problem following the principle of "*forget enough, but no more than necessary.*" At its core, SAUL formulates forgetting as an explicit constraint with a prescribed satisfaction criterion, whereas prior unlearning methods typically specify the desired level of forgetting implicitly through optimization objectives. An augmented Lagrangian controller adaptively adjusts forget-side pressure according to constraint violation and can eventually deactivate the forget-side update as the prescribed criterion remains satisfied. Sharpness-aware updates on both retain and forget objectives, together with a dual-optimizer design that maintains role-separated states, further stabilize the resulting unlearning dynamics. We evaluate SAUL on the TOFU, WMDP, and MUSE benchmarks, demonstrating favorable forgetting–utility trade-offs over representative sharpness- and perturbation-based baselines under benchmark-specific forgetting criteria. Beyond the complete SAUL framework, we further show on TOFU and WMDP that applying the augmented-Lagrangian controller as a *drop-in modifier* to representative baselines improves their post-forgetting utility, demonstrating the practical value of explicit forgetting control.

1 Introduction

Large language models (LLMs) are trained on massive corpora that often contain sensitive, copyrighted, or harmful information (Carlini et al., 2021; Eldan and Russinovich, 2023; Li et al., 2024). **LLM unlearning** has emerged as a critical framework for addressing these concerns by enabling models to "forget" specific data without the prohibitive cost of retraining from scratch (Yao et al., 2024; Liu et al., 2024). A standard unlearning task must erase target knowledge from a *forget set* while preserving general capabilities on a *retain set*. Many existing methods handle this conflict through weighted forget–retain objectives (Zhang et al., 2024; Fan et al., 2025; Bu et al., 2025). However, their trade-off coefficients are sensitive to the dataset, model architecture, and forgetting ratio (Maini et al., 2024; Zhang et al., 2024), making it difficult to explicitly control the desired level of forgetting.

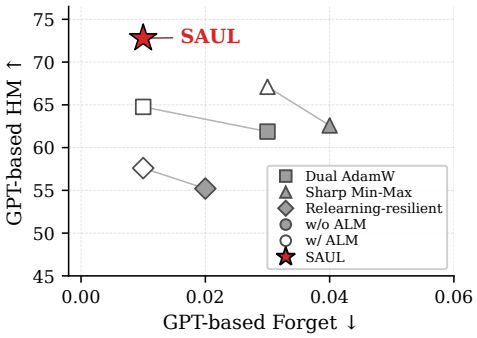


Figure 1: GPT-based utility–forgetting trade-off on ToFU with 1% forget ratio. Better methods lie toward the upper-left, with higher HM and lower Forget. Gray lines connect each baseline to its ALM-enhanced variant. SAUL achieves the highest HM while maintaining a low Forget score.

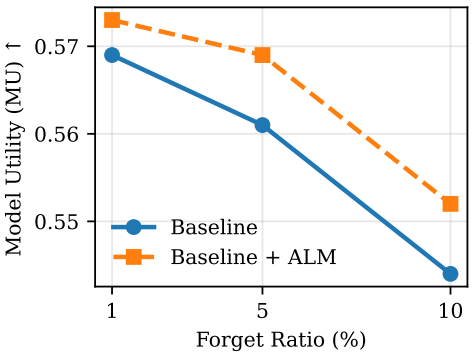


Figure 2: Model utility under an ALM constraint (Dual AdamW baseline). Across all forget ratios, ALM improves model utility over the standard Dual AdamW optimization.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 33202–33230
October 24-29, 2026 ©2026 Association for Computational Linguistics

Recent optimization-based approaches, including bi-level unlearning methods, introduce more structured ways to handle the forget–retain conflict by separating the roles of forgetting and retention across nested or sequential objectives (Reisizadeh et al., 2025; Asif and Amiri, 2025). A parallel line of work mitigates the instability of aggressive forgetting through sharpness- or perturbation-aware optimization (Fan et al., 2025; Tang and Khanna, 2025; Kim et al., 2025; Malekmohammadi et al., 2025), building on Sharpness-Aware Minimization (SAM) (Foret et al., 2021) to obtain solutions stable under local parameter perturbations. While these approaches improve expressiveness and robustness, they still treat forgetting as open-ended *optimization pressure* rather than an explicit constraint with a prescribed satisfaction criterion, allowing unnecessary forgetting to erode retain-side utility.

In this paper, we propose **Sharpness-Aware Augmented Lagrangian Unlearning (SAUL)**, a framework that casts LLM unlearning as a constrained minimization problem following the principle of *forgetting enough, but no more than necessary*. Unlike weighted formulations, SAUL uses a threshold α to specify a target forgetting level, allowing optimization to focus on retain-side utility once the criterion is satisfied.

At the core of SAUL is an *augmented Lagrangian* controller (Hestenes, 1969; Powell, 1969) that adaptively activates or releases forgetting pressure depending on whether the criterion is met. As the prescribed criterion remains satisfied, the projected multiplier decreases and can eventually reach zero and the forget-side gradient is gated off entirely, so optimization focuses solely on preserving retain-side utility—an *adaptive deactivation* mechanism absent from prior constrained or penalty-based unlearning methods. To stabilize the resulting dynamics, SAUL further combines sharpness-aware updates on both retain and forget objectives with a *dual* optimizer design (Zhong et al., 2025) that maintains role-separated states for retention and forgetting, reducing interference between their adaptive moments.

We evaluate SAUL across TOFU, WMDP, and MUSE under matched forgetting criteria, where SAUL improves the forgetting–utility trade-off over representative unlearning baselines. Beyond SAUL itself, Figures 1 and 2 show that the augmented Lagrangian component, applied as a drop-in modifier to existing baselines, consistently improves post-forgetting utility, highlighting the broader value of explicit forgetting control.

Our contributions are summarized as follows:

- We formulate LLM unlearning as constrained minimization, where a target threshold α specifies the desired forgetting level, and propose an augmented-Lagrangian controller that reduces and can deactivate forget-side pressure once the criterion is satisfied.
- We show that the controller can be integrated into representative sharpness- and perturbation-based methods. Across TOFU and WMDP, it consistently improves post-forgetting utility as a drop-in modifier, demonstrating the benefit of explicit forgetting control.
- We instantiate the framework as **SAUL**, integrating the controller with sharpness-aware updates and role-separated dual optimizer states, and achieve favorable forgetting–utility trade-offs across TOFU, WMDP, and MUSE under benchmark-specific forgetting criteria.

2 Related Work

LLM unlearning and weighted objectives. Many LLM unlearning methods formulate forgetting and retention as a scalarized problem, combining forget- and retain-side objectives through a trade-off coefficient. Representative examples include gradient ascent, gradient-difference objectives, loss-adjustment variants, and preference-optimization formulations (Jang et al., 2023; Wang et al., 2024; Bu et al., 2025; Zhang et al., 2024; Mekala et al., 2024). While effective, these methods specify the desired forgetting level implicitly through coefficients that depend on the dataset, model, and forget ratio (Maini et al., 2024; Zhang et al., 2024), potentially causing insufficient forgetting or unnecessary utility degradation.

Bi-level and structured optimization for unlearning. Recent work has explored structured formulations beyond scalarization. BLUR (Reisizadeh et al., 2025) casts unlearning as a bi-level problem whose lower-level objective prioritizes forgetting and upper-level objective preserves retain-side utility. OFMU (Asif and Amiri, 2025) uses an inner forgetting maximization with a similarity-aware penalty and an outer utility-restoration minimization. These methods separate the roles of forgetting and retention, but still lack a prescribed threshold that determines when forgetting is sufficient.

Constrained forgetting and augmented Lagrangian methods. Augmented Lagrangian methods combine dual-variable updates with quadratic penalties to enforce constraints without relying solely on a fixed penalty coefficient (Hestenes, 1969; Bertsekas, 2014; Nocedal and Wright, 2006). Existing constrained LLM unlearning methods place the constraint on the *retain side*: Constrained Entropic Unlearning (Entesari et al., 2025) uses a hard retain constraint, while Cheng et al. (Cheng et al., 2026) constrain the retain-side performance of θ relative to the original model θ_0. In both formulations, the forget-side objective remains active. SAUL instead constrains the *forget side*; as the criterion remains satisfied, the projected multiplier decreases and can eventually reach zero, gating off the forget-side update. We further evaluate the controller as a drop-in modifier to representative sharpness- and perturbation-based objectives on ToFU and WMDP (Section 4.4).

Sharpness-aware and role-separated optimization. Sharpness-Aware Minimization (SAM) promotes solutions stable under local weight perturbations (Foret et al., 2021; Bahri et al., 2022). Sharpness- and perturbation-aware unlearning methods use this principle to stabilize retention or forgetting behavior and reduce susceptibility to relearning-style recovery (Fan et al., 2025; Tang and Khanna, 2025; Kim et al., 2025; Malekmohammadi et al., 2025), but generally lack an explicit criterion for forgetting sufficiency. Separately, dual-optimizer methods reduce interference by maintaining distinct optimizer states for competing objectives (Zhong et al., 2025). SAUL uses these ideas as *supporting components*, while its central contribution is explicit forget-side constraint control.

3 Problem: Unlearning with Constraints

Machine unlearning aims to remove the influence of a designated subset of training data (a *forget set*) from a model, while preserving (and potentially improving) performance on the remaining data (a *retain set*). We view unlearning as a constrained minimization problem. In particular, we consider model parameters $\theta \in \mathbb{R}^d$, an example space $\mathcal{X}$, a label space $\mathcal{Y}$, a *retain (multi-)set* $\mathcal{D}_r \subseteq \mathcal{X} \times \mathcal{Y}$, a *forget (multi-)set* $\mathcal{D}_f \subseteq \mathcal{X} \times \mathcal{Y}$, and a per-example training loss $\ell(\theta, x, y)$ (*e.g.,* cross-entropy). To measure how well the model fits each set, we define the *retain loss* $\mathcal{L}_r(\theta) \coloneqq \frac{1}{|\mathcal{D}_r|} \sum_{(x,y)\in\mathcal{D}_r} \ell(\theta, x, y)$ as the average per-example loss on the retain set, and similarly the *forget loss* $\mathcal{L}_f(\theta) \coloneqq \frac{1}{|\mathcal{D}_f|} \sum_{(x,y)\in\mathcal{D}_f} \ell(\theta, x, y)$ on the forget set. Our goal is to address the following constrained minimization:

$$\min_{\theta} \mathcal{L}_r(\theta) \quad \text{subj. to} \quad \mathcal{L}_f(\theta) \geq \alpha, \tag{1}$$

where α is a user-specified threshold that defines an explicit satisfaction level for the chosen forget-side training measure, rather than specifying the forgetting–retention balance implicitly through a trade-off coefficient. The constraint defines sufficient forgetting in terms of the chosen forget-side loss by requiring $\mathcal{L}_f(\theta) \geq \alpha$, while the retain objective discourages utility degradation beyond what is necessary to satisfy this criterion. Moreover, minimizing the retain loss remains compatible with cases where removing the forget set also improves performance on the remaining data. In Appendix F, we introduce a margin-based certified instantiation in which the threshold α admits a direct prediction-level interpretation, and explain how the resulting certificate can be used post hoc to evaluate models trained with the cross-entropy-based constraint.

4 SAUL: Sharpness-Aware Augmented Lagrangian Unlearning

Building on the constrained formulation in Section 3, we propose *Sharpness-Aware Augmented Lagrangian Unlearning* (**SAUL**), which combines an augmented-Lagrangian constraint with sharpness-aware updates and dual optimizer states to decouple the dynamics of forgetting and retention.

4.1 Augmented Lagrangian Method

Here, we apply the augmented Lagrangian method (ALM) (Hestenes, 1969; Powell, 1969) to unlearning. ALM converts the constrained primal problem (1) into the following unconstrained primal problem with a Lagrangian multiplier:

$$\min_{\theta} \max_{\lambda \geq 0} \mathcal{L}_r(\theta) + \lambda(\alpha - \mathcal{L}_f(\theta)), \tag{2}$$

which is equivalent to (1). However, repeated constraint violations can make the multiplier grow and induce unstable primal updates. ALM can be written as a proximal maximization over the next multiplier estimate λ^+, where the quadratic term prevents the dual variable from moving too far from the current estimate λ. This gives the following

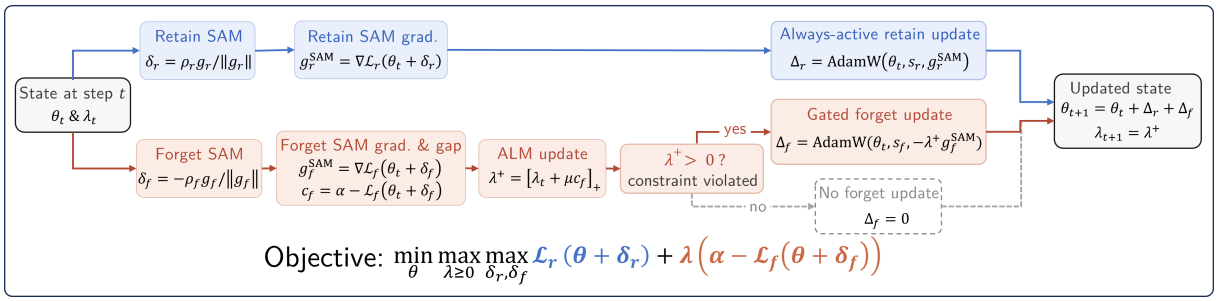


Figure 3: Overview of the SAUL update. The retain branch is always active and applies a SAM-based update to preserve retain-side utility. The forget branch computes a SAM-based forget gradient and an augmented-Lagrangian constraint gap, updates $\lambda^+ = [\lambda_t + \mu c_f]_+$, and applies the forget update only when $\lambda^+ > 0$. Thus, forgetting is activated only when the constraint is violated, while retain-side optimization remains active throughout training.

regularized saddle problem:

$$\min_{\theta} \max_{\lambda^+ \geq 0} \Big[\mathcal{L}_r(\theta) + \lambda^+(\alpha - \mathcal{L}_f(\theta)) - \frac{1}{2\mu}\|\lambda^+ - \lambda\|^2\Big], \tag{3}$$

where $\mu > 0$ is the ALM penalty parameter. Since the objective is concave in λ^+, it admits the closed-form solution:

$$\lambda^+ = \big[\lambda + \mu(\alpha - \mathcal{L}_f(\theta))\big]_+.$$

Let $c_f := \alpha - \mathcal{L}_f(\theta)$ denote the forget-side constraint violation, so that the constraint $\mathcal{L}_f(\theta) \geq \alpha$ is equivalent to $c_f \leq 0$. Plugging the closed-form multiplier update back into (3), the primal update takes the following two-case form:

$$\min_{\theta} \begin{cases} \mathcal{L}_r(\theta) + \lambda c_f + \frac{\mu}{2} c_f^2, & \lambda^+ > 0, \\ \mathcal{L}_r(\theta), & \lambda^+ = 0. \end{cases}$$

When $\lambda^+ > 0$, the augmented term penalizes violations of the forget-side constraint. When the projection yields $\lambda^+ = 0$, the update reduces to retain-loss minimization. For both cases, the updated θ may not satisfy the constraint, so we iterate this procedure by returning to (3) with $\lambda \leftarrow \lambda^+$.

Crucially, as constraint violations disappear, the projected multiplier can decrease and eventually deactivate the forget-side term. When $\lambda^+ = 0$, optimization focuses solely on minimizing the retain loss, avoiding unnecessary forgetting pressure.

4.2 Sharpness-Aware Augmented Lagrangian Method

Sharpness-Aware Minimization (SAM) (Foret et al., 2021) biases optimization toward parameter regions where the loss landscape is flat under local weight perturbations, by approximately solving an inner min–max problem. We extend this idea to both retention and forgetting, so that the resulting solutions remain stable under local weight perturbations on each side.

To this end, we consider a sharpness-aware retain loss

$$\mathcal{L}_r^{\rho_r}(\theta) := \max_{\|\delta_r\| \leq \rho_r} \mathcal{L}_r(\theta + \delta_r)$$

instead of $\mathcal{L}_r(\theta)$. Minimizing this worst-case retain loss yields parameters where the retain landscape is flat, so retain performance remains stable under small weight changes. For the forget side, we consider a sharpness-aware forget loss in the opposite direction:

$$\mathcal{L}_f^{\rho_f}(\theta) := \min_{\|\delta_f\| \leq \rho_f} \mathcal{L}_f(\theta + \delta_f)$$

instead of $\mathcal{L}_f(\theta)$. Here, the inner minimization finds the perturbation *most favorable to recovering forget knowledge*, and maximizing this best-case forget loss forces the forget landscape to remain high even at the most easily recoverable point in the neighborhood. Note the asymmetry: the retain side uses inner max so that the worst case stays low, while the forget side uses inner min so that the best case stays high. The full sharpness-aware augmented Lagrangian method follows ALM in Section 4.1, but replaces $\mathcal{L}_r(\theta)$ and $\mathcal{L}_f(\theta)$ with $\mathcal{L}_r^{\rho_r}(\theta)$ and $\mathcal{L}_f^{\rho_f}(\theta)$, respectively. See Algorithm 1 for our complete algorithm.

Combining the sharpness-aware losses with the ALM formulation in (2) yields the following robust min–max problem:

$$\min_{\theta} \max_{\lambda \geq 0} \max_{\delta_r, \delta_f} \mathcal{L}_r(\theta + \delta_r) + \lambda\big(\alpha - \mathcal{L}_f(\theta + \delta_f)\big). \tag{4}$$

For the forget term, since $\lambda \geq 0$, maximizing $-\lambda \mathcal{L}_f(\theta + \delta_f)$ over δ_f is equivalent to minimizing

$\mathcal{L}_f(\theta + \delta_f)$, which matches the sharpness-aware forget loss. Because $\delta_r = 0$ and $\delta_f = 0$ are feasible perturbations, (4) upper-bounds the ALM objective in (2) for any fixed θ and λ. In practice, as in standard SAM, we approximate each inner problem with a single first-order perturbation step. Minimizing this upper bound therefore controls the original ALM objective while encouraging stability under local weight perturbations on both sides. This gives a principled interpretation of the forget-side sharpness step as approximating a robust forgetting constraint that should remain satisfied even under local weight perturbations. We emphasize that this robustness is restricted to weight-space perturbations and does not by itself imply resistance to prompt-level adversarial attacks or recovery attacks.

4.3 Dual Optimizer States

While the augmented Lagrangian formulation in Section 4.1 specifies when the forgetting constraint should be enforced through the multiplier λ, and the sharpness-aware losses in Section 4.2 improve the *robustness* of the retain/forget objectives, a remaining source of instability lies in the *optimizer dynamics* used to carry out these updates (Zhong et al., 2025).

A common implementation optimizes the (sharpness-aware) ALM objective with a *single* adaptive optimizer (e.g., AdamW), which couples forgetting and retention through shared first- and second-moment estimates. In unlearning, however, retain-side and forget-side gradients often have conflicting semantics and markedly different scales—utility-preserving descent versus constraint-enforcing updates—and mixing such heterogeneous signals in one state can lead to optimizer-state interference, where accumulated moments become inconsistent with either role.

To mitigate this, SAUL maintains *two independent optimizer states* for the *same* model parameters θ: a forgetting optimizer state s_f and a retention optimizer state s_r (both based on AdamW in our implementation). The parameters remain shared; only the optimizer statistics are decoupled. s_f is updated using forget-side gradients only, while s_r is updated using retain-side gradients only, so each state accumulates objective-consistent moments suitable for its role. At each iteration, SAUL first computes both g_r^{SAM} and g_f^{SAM} at the current parameters θ_t. It then applies the forget-side update only when $\lambda^+ > 0$, followed by the retain-side update using role-separated optimizer states. Algorithm 1 summarizes the complete procedure, combining (i) ALM-based adaptive activation of forgetting, (ii) sharpness-aware gradients for robust optimization, and (iii) role-separated optimizer states for stable unlearning dynamics.

4.4 ALM-Augmented Variants

Beyond SAUL itself, the augmented-Lagrangian controller can also be applied as a drop-in modifier to existing unlearning objectives. Given a baseline objective $\mathcal{J}(\theta)$ and a forgetting measure $F(\theta)$, where larger $F(\theta)$ indicates stronger forgetting, we impose the constrained form

$$\min_{\theta} \mathcal{J}(\theta) \quad \text{s.t.} \quad F(\theta) \geq \alpha.$$

ALM introduces a dual variable $\lambda \geq 0$ and a penalty parameter $\mu > 0$, with the projected update

$$\lambda^+ \leftarrow \big[\lambda + \mu(\alpha - F(\theta))\big]_+.$$

This update increases forget-side pressure when the constraint is violated and reduces it as the constraint remains satisfied, replacing fixed trade-off tuning with constraint-driven adaptation.

Unlike SAUL, which integrates ALM with sharpness-aware updates on both retain and forget sides plus role-separated optimizer states, these ALM-augmented baselines apply only the constraint mechanism on top of the original baseline structure. These ALM-augmented variants allow us to isolate the effect of constraint-driven forgetting control from the underlying sharpness-aware objectives. We evaluate the post-forgetting utility of these ALM-augmented variants in Section 5, with detailed baseline-specific objectives provided in Appendix A.3.

5 Experiments

We evaluate SAUL on three LLM unlearning benchmarks—ToFU (Maini et al., 2024), WMDP (Li et al., 2024), and MUSE (Shi et al., 2024)—covering author-specific fictitious forgetting, hazardous knowledge removal, and verbatim memorization. All methods are tuned under matched forgetting criteria so that comparisons reflect the forgetting–utility trade-off at comparable levels of forgetting. Detailed training configurations and hyperparameters are provided in Appendix B.

5.1 Setup

Benchmarks. We evaluate methods across three benchmark-specific settings, following the established model and evaluation configurations

Table 1: Unlearning performance with forget ratio = 1%. All methods are tuned to achieve Forget ROUGE close to 0.03 when possible. **Bold** and underlined values denote the best and second-best results for each metric, respectively. Red values indicate cases where unlearning is not sufficiently achieved; methods with red Forget values are excluded from the best/second-best ranking. A dash (–) indicates that the metric is not applicable under our hyperparameter tuning protocol.

Method	Automatic		GPT-based				
	MU↑	F_{ROUGE} ↓	Retain↑	World Facts↑	Real Authors↑	HM ↑	Forget↓
Original Model	0.60	0.87	90.0	80.3	81.4	81.9	82.5
Single AdamW	0.45 ± 0.00	0.02 ± 0.03	50.45 ± 0.72	70.43 ± 0.97	35.20 ± 2.39	48.01 ± 1.50	0.17 ± 0.08
Dual AdamW	0.57 ± 0.00	0.02 ± 0.00	64.25 ± 0.56	80.00 ± 1.43	49.00 ± 1.58	61.87 ± 0.87	0.03 ± 0.02
Dual AdamW + ALM	0.57 ± 0.00	**0.01 ± 0.00**	63.20 ± 0.27	**82.56 ± 1.43**	54.40 ± 1.14	64.76 ± 0.68	**0.01 ± 0.01**
Sharp Min–Max	0.56 ± 0.00	0.02 ± 0.00	53.45 ± 0.37	81.54 ± 0.97	59.00 ± 1.58	62.59 ± 0.58	0.04 ± 0.02
Sharp Min–Max + ALM	0.58 ± 0.01	**0.01 ± 0.00**	58.05 ± 0.37	80.85 ± 0.76	66.20 ± 0.84	67.11 ± 0.25	0.03 ± 0.03
Relearning-resilient	0.51 ± 0.01	**0.01 ± 0.00**	46.40 ± 0.38	77.78 ± 0.60	50.20 ± 0.84	55.21 ± 0.23	0.02 ± 0.02
Relearning-resilient + ALM	0.52 ± 0.01	0.02 ± 0.01	49.90 ± 0.29	80.00 ± 1.43	51.20 ± 0.84	57.60 ± 0.15	**0.01 ± 0.01**
BLUR	0.57 ± 0.14	0.28 ± 0.20	63.00 ± 1.40	80.00 ± 3.40	73.60 ± 4.00	71.40 ± 2.50	36.00 ± 2.80
PDU	0.56 ± 0.02	0.17 ± 0.02	60.42 ± 1.04	79.2 ± 0.99	62.00 ± 2.00	66.20 ± 0.70	8.33 ± 3.82
SAUL	**0.59 ± 0.01**	0.02 ± 0.02	**68.25 ± 1.09**	79.66 ± 1.27	**71.40 ± 0.55**	**72.79 ± 0.54**	**0.01 ± 0.01**
w/o ALM	0.57 ± 0.01	**0.01 ± 0.01**	64.95 ± 0.60	78.29 ± 1.55	41.40 ± 1.14	57.32 ± 0.48	0.02 ± 0.02
w/o SAM	0.55 ± 0.01	**0.01 ± 0.01**	64.50 ± 0.68	82.39 ± 1.30	56.20 ± 1.92	66.01 ± 1.01	0.02 ± 0.01
w/o ALM + SAM	0.57 ± 0.00	**0.01 ± 0.01**	64.10 ± 1.11	80.51 ± 1.64	47.60 ± 1.14	61.17 ± 0.39	0.16 ± 0.33

used for each benchmark rather than adopting a single shared backbone. For **ToFU**, we use `Llama-3.2-1B-Instruct` as the main backbone under the standard question–answering setting with forgetting ratios of 1%, 5%, and 10%. We also evaluate GPT-paraphrased questions to test robustness to query rephrasing; their construction is described in Appendix C, and the corresponding results are reported in Appendix D. Additional `Llama-3.2-3B-Instruct` results are reported in Appendix D. For **WMDP**, we follow its benchmark-specific setup using `Zephyr-7B-`β and unlearn WMDP-Bio and WMDP-Cyber, evaluating hazardous-knowledge accuracy and MMLU utility. For **MUSE**, we follow the benchmark-specific `Llama-2-7B` setup and use the standard VerbMem, KnowMem, and PrivLeak evaluation protocol. We report MUSE Books as the main evaluation setting and provide additional results on MUSE News in Appendix D.2.

Baselines. We compare against representative sharpness- and perturbation-based unlearning methods: the relearning-resilient sharpness-aware baseline of Fan et al. (Fan et al., 2025) and Sharp Min–Max (Tang and Khanna, 2025) without masking. For brevity, we refer to the former as Relearning-resilient in tables and result discussions. We also include Single AdamW, Dual AdamW (Zhong et al., 2025), and BLUR (Reisizadeh et al., 2025). On ToFU, we additionally compare against Primal-Dual Unlearning (PDU) (Entesari et al., 2025), a recent retain-side constrained method that optimizes forgetting subject to an explicit retain-utility constraint. For baselines that admit a comparable operating point, we further evaluate a "+ ALM" variant to isolate the effect of explicit forgetting control. Detailed baseline definitions and implementation details are provided in Appendix A.

5.2 Results on ToFU

Evaluation metrics. We assess unlearning performance using two complementary automatic metrics from ToFU: *model utility* (MU), which aggregates the probability assigned to the ground-truth answer, ROUGE-L recall, and the truth ratio via the harmonic mean; and *Forget ROUGE* (F_{ROUGE}), which measures ROUGE-L recall between generated and ground-truth answers on the forget set. We additionally employ a GPT-based semantic evaluation that produces binary entailment judgments between generated and reference answers, applied to four subsets (*Forget*, *Retain*, *Real Authors*, *World Facts*); we report the harmonic mean (HM) of GPT-based scores on the latter three to summarize utility beyond forgetting.

Main results. Table 1 reports results at forget ratio 1%. Since SAUL formulates forgetting as a constraint, we compare methods under the matched-forgetting criterion $F_{\text{ROUGE}} \leq 0.03$. SAUL satisfies this criterion with Forget ROUGE of 0.02 and achieves the best harmonic mean (72.79), outperforming the next-best constraint-satisfying method

Table 2: Unlearning performance with forget ratio = 1% under GPT-paraphrased questions. All hyperparameters are selected on the original-question setting and then reused for paraphrased-question evaluation without additional tuning. Notation follows Table 1.

Method	Automatic		GPT-based				
	MU↑	F_{ROUGE} ↓	Retain↑	World Facts↑	Real Authors↑	HM ↑	Forget↓
Original Model	0.54	0.44	56.75	79.49	77.00	69.46	50.0
Single AdamW	0.49 ± 0.01	0.03 ± 0.01	46.46 ± 2.38	68.22 ± 9.94	25.60 ± 11.22	38.25 ± 8.79	2.00 ± 2.09
Dual AdamW	0.51 ± 0.02	0.03 ± 0.01	**53.14 ± 2.09**	71.98 ± 7.90	47.20 ± 17.78	55.89 ± 16.37	**0.50 ± 1.12**
Dual AdamW + ALM	**0.53 ± 0.01**	0.04 ± 0.00	52.62 ± 1.87	75.72 ± 5.01	55.40 ± 10.60	59.20 ± 5.36	2.50 ± 0.00
Sharp Min–Max	0.47 ± 0.04	0.12 ± 0.18	43.12 ± 3.51	70.10 ± 5.16	41.80 ± 5.93	48.74 ± 4.35	1.50 ± 2.24
Sharp Min–Max + ALM	0.50 ± 0.03	0.05 ± 0.02	49.22 ± 1.74	73.86 ± 3.45	51.60 ± 8.50	56.14 ± 4.46	1.50 ± 1.37
Relearning-resilient	-	-	-	-	-	-	-
Relearning-resilient + ALM	-	-	-	-	-	-	-
BLUR	0.53 ± 0.03	0.44 ± 0.21	51.90 ± 3.30	78.80 ± 3.10	72.80 ± 4.50	65.60 ± 3.70	49.00 ± 4.50
PDU	0.50 ± 0.01	0.34 ± 0.02	42.83 ± 2.36	74.93 ± 1.31	59.67 ± 1.15	56.09 ± 1.33	36.67 ± 2.89
SAUL	**0.53 ± 0.01**	0.13 ± 0.22	52.28 ± 1.40	**78.44 ± 2.29**	**63.60 ± 6.23**	**62.90 ± 2.18**	1.50 ± 1.37
w/o ALM	0.50 ± 0.03	0.03 ± 0.02	52.22 ± 2.53	70.94 ± 9.91	46.80 ± 24.77	51.44 ± 15.83	1.00 ± 2.24
w/o SAM	**0.53 ± 0.01**	0.03 ± 0.01	53.12 ± 1.50	75.23 ± 4.47	62.60 ± 12.99	61.65 ± 4.73	1.00 ± 1.37
w/o ALM + SAM	0.50 ± 0.02	**0.02 ± 0.01**	52.12 ± 1.62	68.02 ± 10.70	39.80 ± 16.30	49.05 ± 11.33	**0.50 ± 1.12**

(Sharp Min–Max + ALM, HM 67.11) by 5.7 points.

Although BLUR attains a similar HM (71.40), its Forget ROUGE is 0.28. Our reproduction matches BLUR's reported Forget Quality, but answer-level evaluation still reveals substantial forget-set leakage. We therefore include BLUR for completeness, while interpreting it as outside the strict answer-level forgetting regime targeted by SAUL.

Effect of ALM on Utility and Generalization. Figures 1 and 2 show that adding ALM to baseline methods improves both model utility and GPT-based evaluation metrics. While improvements on the *Retain* set are moderate, the gains are larger on the *Real Authors* subset, which evaluates generalization to neighborhood data. This pattern aligns with SAUL's design principle of forgetting only as much as necessary: once the criterion is met, ALM deactivates forget-side pressure and avoids unnecessary updates that erode semantically related knowledge in neighborhood subsets, whereas open-ended forgetting objectives continue applying forget-side gradients and degrade such data more aggressively.

We further evaluate this behavior under paraphrased questions, where forget queries preserve the original semantics but vary in lexical and structural form. As shown in Table 2, SAUL remains effective under this query perturbation, suggesting that its forgetting behavior is not tied to the surface form of the original questions. Additional paraphrased-question results for other forgetting ratios and larger backbones are provided in Appendix D, where the same overall trend holds.

Efficacy and Constraint Satisfaction. As shown in Table 1, SAUL achieves strong overall performance while satisfying the forgetting constraint. Adding ALM improves model utility over the corresponding non-ALM variants, indicating that the constraint-based formulation can enforce sufficient forgetting without excess retention degradation.

This trend holds beyond the main setting. In Appendix D, we report additional results for 5% and 10% forget ratios, paraphrased-question evaluation, and the larger LLaMA-3.2-3B backbone. Across these settings, SAUL remains competitive and often achieves the best harmonic mean, showing that the proposed ALM–SAM formulation generalizes across forgetting ratios, query perturbations, and model scales.

Ablation Study To separate the contributions of sharpness-aware optimization and the augmented Lagrangian formulation, we evaluate three SAUL variants: (i) removing ALM while keeping SAM, (ii) removing SAM while keeping ALM, and (iii) removing both. In full SAUL, ALM is applied to the sharpness-aware forget loss evaluated at the local perturbation most favorable to recovering the forgotten knowledge, thereby enforcing the forgetting constraint over a local weight-space neighborhood rather than only at the current parameters. In contrast, the baseline "+ ALM" variants and the SAUL "w/o SAM" variant apply ALM to the clean forget loss, allowing us to isolate the effect of constraint enforcement on sharpness-aware and clean forget measures.

5.3 Results on WMDP

Table 3: Unlearning performance on WMDP using `Zephyr-7B-`β. WMDP-Bio and WMDP-Cyber measure residual hazardous knowledge on the forget domains (lower is better), while MMLU measures general utility preservation (higher is better).

Method	Bio↓	Cyber↓	MMLU↑
Original	0.637	0.440	0.581
Sharp Min–Max	0.260 $\pm$0.012	0.260 $\pm$0.004	0.535 $\pm$0.002
Sharp Min–Max + ALM	0.261 $\pm$0.010	0.256 $\pm$0.007	**0.550** $\pm$0.003
Relearning-resilient	0.262 $\pm$0.000	0.262 $\pm$0.000	0.414 $\pm$0.000
Relearning-resilient + ALM	0.255 $\pm$0.000	0.260 $\pm$0.000	0.424 $\pm$0.000
BLUR	0.262 $\pm$0.010	0.253 $\pm$0.009	0.540 $\pm$0.004
BLUR + ALM	0.265 $\pm$0.011	0.255 $\pm$0.011	0.543 $\pm$0.003
SAUL	0.267 $\pm$0.005	0.256 $\pm$0.005	0.544 $\pm$0.003

Table 3 reports results on WMDP. All compared methods are evaluated at comparable forgetting levels, with the average accuracy over WMDP-Bio and WMDP-Cyber targeted around 0.26, close to the random-chance accuracy of 0.25 for four-way multiple-choice questions. Under this matched-forgetting condition, adding the augmented-Lagrangian controller consistently improves retain-side utility across different baseline methods. In particular, MMLU accuracy increases from 0.535 to 0.550 for Sharp Min–Max, from 0.414 to 0.424 for Relearning-resilient, and from 0.540 to 0.543 for BLUR, while their WMDP accuracies remain at comparable levels. These results indicate that the ALM controller is not specific to SAUL and can improve the forgetting–utility trade-off of existing unlearning objectives by regulating forget-side pressure. SAUL achieves an MMLU accuracy of 0.544 ± 0.003 under a similarly matched forgetting level, remaining competitive with the strongest baseline variants.

5.4 Results on MUSE Books

Table 4: Unlearning performance on MUSE Books using `Llama-2-7B`. VerbMem and KnowMem on $\mathcal{D}_f$ measure residual memorization on the forget set (lower is better); PrivLeak measures privacy leakage (closer to 0 is better); KnowMem on $\mathcal{D}_r$ measures retain-side knowledge preservation (higher is better).

Method	VerbMem ↓	KnowMem $\mathcal{D}_f$ ↓	PrivLeak → 0	KnowMem $\mathcal{D}_r$ ↑
Original	99.8	59.4	-57.5	66.9
Retrain	14.3	28.9	0.0	74.5
Sharp Min–Max	0.00 ± 0.00	5.37 ± 1.29	-33.53 ± 0.86	39.76 ± 1.82
Relearning-resilient	0.03 ± 0.06	5.67 ± 0.77	-2.13 ± 3.89	39.03 ± 1.03
BLUR	0.00 ± 0.00	3.09 ± 4.97	-30.77 ± 5.41	**48.40 ± 1.50**
SAUL	**0.00 ± 0.00**	0.77 ± 1.34	**-16.66 ± 2.90**	48.25 ± 5.82

Table 4 reports results on MUSE Books, which evaluates the removal of verbatim and factual memorization from the Harry Potter corpus. PrivLeak measures similarity to the retrained reference distribution, with values closer to 0 indicating better alignment. For the direct comparison, BLUR and SAUL are tuned toward the same knowledge-forgetting target, while the remaining baselines are reported at their default operating points.

Compared with BLUR, SAUL achieves the same zero VerbMem (0.00 ± 0.00) while obtaining a PrivLeak score substantially closer to zero (-16.66 ± 2.90 vs. -30.77 ± 5.41). At the same time, SAUL preserves a comparable level of retain-side knowledge, with KnowMem on $\mathcal{D}_r$ of 48.25 ± 5.82 compared with 48.40 ± 1.50 for BLUR. These results show that SAUL reduces privacy leakage without sacrificing retain-side knowledge at a comparable forgetting operating point.

The remaining baselines exhibit different trade-offs. Relearning-resilient achieves the PrivLeak score closest to the retrained reference (-2.13 ± 3.89), but preserves substantially less retain-side knowledge than SAUL (39.03 ± 1.03 vs. 48.25 ± 5.82). Sharp Min–Max also achieves zero VerbMem, but incurs greater retain-side degradation and obtains a PrivLeak score farther from zero than SAUL. Additional results on the MUSE News benchmark are provided in Appendix D.2.

6 Conclusion

We propose *Sharpness-Aware Augmented Lagrangian Unlearning (SAUL)*, which formulates LLM unlearning as constrained optimization under the principle of "*forget enough, but no more than necessary.*" SAUL uses an augmented-Lagrangian controller to reduce forget-side pressure once a prescribed forgetting criterion is met, and combines it with sharpness-aware updates and role-separated optimizer states for stable unlearning. Experiments on ToFU, WMDP, and MUSE Books show that SAUL improves the forgetting–utility trade-off under matched forgetting criteria, preserving retain-side utility, neighborhood-data performance, and general capability while avoiding excessive forgetting. We further show that the augmented-Lagrangian controller can serve as a method-agnostic drop-in modifier for representative sharpness- and perturbation-based baselines, highlighting the broader value of explicit forgetting control in LLM unlearning.

7 Limitations

SAUL requires selecting a forgetting threshold that specifies when the target knowledge is sufficiently removed. Although this threshold provides an interpretable interface for controlling the forgetting–utility trade-off, its optimal value may depend on the dataset, model scale, forget ratio, and evaluation metric. Developing automatic or theoretically grounded threshold-selection procedures remains an important direction for future work.

Our evaluation focuses on benchmark-level unlearning across ToFU, WMDP, and MUSE. While we include paraphrased-question evaluation to test robustness to surface-form changes, we do not provide formal guarantees against all adversarial prompting, relearning, or recovery attacks. Future work should study stronger adaptive attacks and broader forms of post-unlearning robustness.

Finally, SAUL introduces additional computation and memory primarily from sharpness-aware perturbation steps and role-separated optimizer states, while the augmented-Lagrangian controller itself remains lightweight. Although the overhead is moderate in our experiments, scaling the method to substantially larger models or more complex unlearning targets may require further memory- and compute-efficient implementations; further details are provided in Appendix E.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2019-II191906, Artificial Intelligence Graduate School Program (POSTECH) (5%); RS-2024-00457882, National AI Research Lab Project (10%); No. RS-2024-00509258 and No. RS-2024-00469482, Global AI Frontier Lab (40%); RS-2025-00560062 (45%)).

References

Sadia Asif and Mohammad Mohammadi Amiri. 2025. Ofmu: Optimization-driven framework for machine unlearning. *Preprint*, arXiv:2509.22483.

Dara Bahri, Hossein Mobahi, and Yi Tay. 2022. Sharpness-aware minimization improves language model generalization. *Preprint*, arXiv:2110.08529.

Dimitri P Bertsekas. 2014. *Constrained optimization and Lagrange multiplier methods*. Academic press.

Zhiqi Bu, Xiaomeng Jin, Bhanukiran Vinzamuri, Anil Ramakrishna, Kai-Wei Chang, Volkan Cevher, and Mingyi Hong. 2025. Unlearning as multi-task optimization: A normalized gradient difference approach with an adaptive learning rate. *Preprint*, arXiv:2410.22086.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, and 1 others. 2021. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

Jingpu Cheng, Ping Liu, Qianxiao Li, and Chi Zhang. 2026. Machine unlearning under retain-forget entanglement. *Preprint*, arXiv:2603.26569.

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

Taha Entesari, Arman Hatami, Rinat Khaziev, Anil Ramakrishna, and Mahyar Fazlyab. 2025. Constrained entropic unlearning: A primal-dual framework for large language models. *Preprint*, arXiv:2506.05314.

Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. 2025. Towards llm unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. *Preprint*, arXiv:2502.05374.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2021. Sharpness-aware minimization for efficiently improving generalization. *Preprint*, arXiv:2010.01412.

Magnus R. Hestenes. 1969. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320.

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, Toronto, Canada. Association for Computational Linguistics.

Hoki Kim, Keonwoo Kim, Sungwon Chae, and Sangwon Yoon. 2025. Unlearning-aware minimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, and 27 others. 2024. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Proceedings of the 41st International*

Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, pages 28525–28550. PMLR.

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. 2024. Rethinking machine unlearning for large language models. *Preprint*, arXiv:2402.08787.

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

Saber Malekmohammadi, Hong kyu Lee, and Li Xiong. 2025. Sharpness-aware parameter selection for machine unlearning. *Preprint*, arXiv:2504.06398.

Anmol Mekala, Vineeth Dorna, Shreya Dubey, Abhishek Lalwani, David Koleczek, Mukund Rungta, Sadid Hasan, and Elita Lobo. 2024. Alternate preference optimization for unlearning factual knowledge in large language models. *Preprint*, arXiv:2409.13474.

Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization*, 2 edition. Springer.

Michael JD Powell. 1969. A method for nonlinear constraintsa in minimization problems. *Optimization*, pages 283–298.

Hadi Reisizadeh, Jinghan Jia, Zhiqi Bu, Bhanukiran Vinzamuri, Anil Ramakrishna, Kai-Wei Chang, Volkan Cevher, Sijia Liu, and Mingyi Hong. 2025. Blur: A bi-level optimization approach for llm unlearning. *Preprint*, arXiv:2506.08164.

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

Haoran Tang and Rajiv Khanna. 2025. Sharpness-aware machine unlearning. *Preprint*, arXiv:2506.13715.

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

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024. Large language model unlearning. *Preprint*, arXiv:2310.10683.

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

Xuyang Zhong, Haochen Luo, and Chen Liu. 2025. Dualoptim: Enhancing efficacy and stability in machine unlearning with dual optimizers. *Preprint*, arXiv:2504.15827.

A Baselines

We compare SAUL against standard optimization baselines, recent LLM unlearning methods, and ALM-augmented variants of representative sharpness-based baselines. All methods are evaluated under the same backbone, dataset split, training budget, and evaluation metrics.

A.1 Standard Optimization Baselines

Single AdamW. Single AdamW optimizes the unlearning objective with a single AdamW optimizer state shared across retain- and forget-side updates. This baseline serves as the simplest reference for assessing whether the proposed role-separated optimizer design improves stability beyond conventional adaptive optimization.

Dual AdamW. Dual AdamW maintains separate AdamW optimizer states for retain- and forget-side updates while sharing the same model parameters. Because retain and forget objectives can induce gradients with different semantics and scales, separating their adaptive statistics can reduce optimizer-state interference. This baseline isolates the effect of role-separated optimizer states without the full constrained sharpness-aware formulation.

A.2 Recent LLM Unlearning Baselines

Relearning-resilient Unlearning. Relearning-resilient Unlearning improves resistance to relearning-style recovery by applying sharpness-aware optimization to the forget-side objective (Fan et al., 2025). Its objective robustifies forgetting through an inner maximization over local weight perturbations, while the retain side enters as a weighted regularization term. This baseline is relevant because it directly tests whether forget-side sharpness alone is sufficient, or whether explicit forgetting control and retain-side sharpness-aware updates provide additional benefits.

Sharp Min–Max. Sharp Min–Max applies different sharpness-aware dynamics to the retain and forget sides (Tang and Khanna, 2025). The retain side minimizes worst-case retain loss to encourage flatness and preserve utility, while the forget side promotes sharpness-increasing dynamics to amplify forgetting. We use the no-masking variant in our experiments. This baseline is closely related to SAUL because it already separates retain- and forget-side perturbation roles, but it does not adaptively deactivate forget-side pressure based on a prescribed forgetting threshold.

BLUR. BLUR is a bi-level optimization method for LLM unlearning (Reisizadeh et al., 2025). It is closely related to SAUL because it also departs from a simple weighted-sum formulation. However, while BLUR structures unlearning through a bi-level optimization problem, SAUL formulates forgetting as an explicit constraint and optimizes retain-side utility subject to satisfying a target forgetting level. This comparison evaluates the difference between bi-level trade-off optimization and forget-constrained satisficing.

Primal-Dual Unlearning (PDU). PDU is a constrained optimization method for LLM unlearning that minimizes a forget-side objective subject to an explicit retain-side utility constraint (Entesari et al., 2025). It is closely related to SAUL because both methods use primal–dual optimization to control the forgetting–utility trade-off through an explicit constraint. However, PDU constrains retain-side degradation while leaving forgetting as the primary optimization objective, whereas SAUL constrains the forget side and minimizes retain loss subject to achieving a prescribed forgetting level. This comparison evaluates the difference between retain-constrained forgetting and forget-constrained utility preservation.

A.3 ALM-Augmented Sharpness-Based Variants

To test whether the augmented-Lagrangian controller is useful beyond SAUL itself, we also apply it as a drop-in modifier to representative sharpness- and perturbation-based baselines. Given a baseline objective $\mathcal{J}(\theta)$ and a forgetting measure $F(\theta)$ where larger values indicate stronger forgetting, we impose

$$\min_{\theta} \mathcal{J}(\theta) \quad \text{s.t.} \quad F(\theta) \geq \alpha.$$

Method	Core objective $\mathcal{J}(\theta)$	Where sharpness/ perturbation enters
Relearning-resilient baseline	$\min_{\theta} \Big[\max_{\|\delta\|_p \le \rho} \mathcal{L}_f^{\mathrm{NPO}}(\theta+\delta) + \lambda\, \mathcal{L}_r(\theta) \Big]$	**Forget-side perturbation:** the inner maximization over weight perturbations is applied to the forget-side objective, while the retain side remains an unperturbed weighted term.
Sharp Min–Max (no masking)	Retain: $\min_{\theta} \max_{\|\delta_r\| \le \rho_r} \mathcal{L}_r(\theta+\delta_r)$ Forget: $\min_{\theta} \Big\{ \mathcal{L}_f(\theta) - \big[\max_{\|\delta_f\| \le \rho_f} \mathcal{L}_f(\theta+\delta_f) - \mathcal{L}_f(\theta) \big] \Big\}$	**Two-role perturbation dynamics:** the retain side minimizes worst-case retain loss for utility preservation, while the forget side promotes sharpness-increasing dynamics to amplify forgetting.

Table 5: Summary of representative sharpness- and perturbation-based unlearning objectives considered for ALM augmentation. $\mathcal{L}_r$ and $\mathcal{L}_f$ denote retain and forget losses; ρ, ρ_r, ρ_f are weight-space perturbation radii; λ denotes a trade-off coefficient in the original baseline objective.

The projected multiplier update is

$$\lambda^{+} \leftarrow \big[\lambda + \mu(\alpha - F(\theta))\big]_{+}.$$

The multiplier increases forget-side pressure when the constraint is violated and reduces it as the constraint remains satisfied.

ALM-augmented Relearning-resilient baseline. For Relearning-resilient Unlearning (Fan et al., 2025), the original baseline applies sharpness-aware optimization to the forget-side objective while using $\lambda\mathcal{L}_r(\theta)$ as a retain-side weighted term. We define the forgetting measure as

$$F_{\mathrm{relearn}}(\theta) \triangleq \max_{\|\delta\| \le \rho} \mathcal{L}_f^{\mathrm{NPO}}(\theta+\delta).$$

Enforcing $F_{\mathrm{relearn}}(\theta) \ge \alpha$ requires the forget-side objective to remain large under local weight perturbations. The fixed trade-off coefficient is then replaced by the adaptive multiplier induced by the augmented-Lagrangian controller.

ALM-augmented Sharp Min–Max. For Sharp Min–Max (Tang and Khanna, 2025), the retain-side update minimizes worst-case retain loss, while the forget-side update promotes sharpness-increasing dynamics. We apply ALM at the update level: the retain-side SAM update is kept intact, and the multiplier activates and scales the forget-side update when the forgetting constraint is violated. As $F(\theta)$ remains above α, the projected multiplier can decrease toward zero, reducing or eventually skipping the forget-side step so that optimization focuses on retain-side sharpness minimization. Compared with these ALM-augmented baselines, SAUL combines three design choices in a single method: explicit forget-side constraint control, sharpness-aware updates on both retain and forget objectives, and role-separated optimizer states.

Implementation and Evaluation. We tune method-specific hyperparameters using a comparable validation protocol and follow the original recommended settings when available. For all methods, we select operating points under matched forgetting criteria so that comparisons reflect utility differences at comparable levels of forgetting. SAUL is implemented following Algorithm 1, which combines sharpness-aware updates, augmented-Lagrangian multiplier control, and separate optimizer states for the retain- and forget-side objectives.

Algorithm 1 Sharpness-aware Augmented Lagrangian Method (SAUL)

Require: Initial parameters θ, multiplier $\lambda \geq 0$, optimizer states s_r, s_f; retain batch $\mathcal{B}_r$, forget batch $\mathcal{B}_f$; SAM radii ρ_r, ρ_f; ALM penalty $\mu > 0$; threshold α; small $\varepsilon_{\text{pert}}$.

procedure $\mathrm{SAUL}(\theta, \lambda, s_r, s_f, \mathcal{B}_r, \mathcal{B}_f, \rho_r, \rho_f, \mu, \alpha, \varepsilon_{\text{pert}})$
for $t = 1, 2, \ldots, T$ **do**
  $(g_r^{\text{SAM}}, \delta_r^*) \leftarrow \textsc{Grad}_{\text{SAM}}(\theta, \mathcal{B}_r, \mathcal{L}_r, \rho_r, \texttt{retain})$
  $(g_f^{\text{SAM}}, \delta_f^*) \leftarrow \textsc{Grad}_{\text{SAM}}(\theta, \mathcal{B}_f, \mathcal{L}_f, \rho_f, \texttt{forget})$
  $c_f \leftarrow \alpha - \mathcal{L}_f(\theta + \delta_f^*; \mathcal{B}_f)$
  $\lambda^+ \leftarrow \left[\lambda + \mu\, c_f\right]_+$
  if $\lambda^+ > 0$ **then**
    $\tilde{g}_f \leftarrow -\lambda^+ g_f^{\text{SAM}}$
    $(\theta, s_f) \leftarrow \textsc{AdamW}(\theta, s_f, \tilde{g}_f)$
  end if
  $(\theta, s_r) \leftarrow \textsc{AdamW}(\theta, s_r, g_r^{\text{SAM}})$
  $\lambda \leftarrow \lambda^+$
end for
return $\theta, \lambda, s_r, s_f$

procedure $\textsc{Grad}_{\text{SAM}}(\theta, \mathcal{B}, \mathcal{L}, \rho, \texttt{mode})$
  $g \leftarrow \nabla_\theta \mathcal{L}(\theta; \mathcal{B})$
  if $\texttt{mode} = \texttt{retain}$ **then** $\delta^\star \leftarrow \rho \dfrac{g}{\|g\|_2 + \varepsilon_{\text{pert}}}$
  else $\delta^\star \leftarrow -\rho \dfrac{g}{\|g\|_2 + \varepsilon_{\text{pert}}}$
  $g^{\text{SAM}} \leftarrow \nabla_\theta \mathcal{L}(\theta + \delta^\star; \mathcal{B})$
  return $(g^{\text{SAM}}, \delta^\star)$
end procedure

B Experimental Setup

B.1 Training Configuration

For TOFU, models are trained for 10 epochs with a batch size of 8. For WMDP, we follow the RMU training setup and train for one epoch over at most 150 mini-batches with a batch size of 4, using two GPUs to host the updated and frozen models. For MUSE-Books, we train for 10 epochs with a per-device batch size of 1, using gradient checkpointing when needed for memory efficiency. Our method uses AdamW-style parameter updates, maintaining separate optimizer states for the forget and retain objectives unless otherwise specified by a baseline. Experiments are conducted on NVIDIA RTX 6000/A6000-class GPUs for TOFU and WMDP, and on H200 and RTX Pro 6000-class GPUs for MUSE-Books. For TOFU and WMDP, we report mean and standard deviation over five independent runs. For MUSE-Books, we report results from a single run due to the substantially higher computational cost of 7B-scale memorization evaluation.

B.2 Hyperparameter Selection

Hyperparameters for all methods are selected via grid search.

Table 6: Final hyperparameter configurations for the main ToFU setting with forget ratio $= 1\%$.

Method	Learning rate	SAM (ρ_f, ρ_r)	Risk α	λ LR
SAUL	1×10^{-5}	1×10^{-3}	10	1×10^{-3}
w/o ALM	1×10^{-5}	1×10^{-3}	–	–
w/o SAM	1×10^{-5}	$(-, 1 \times 10^{-3})$	30	1×10^{-4}
w/o ALM + SAM	1×10^{-5}	$(-, 1 \times 10^{-3})$	–	–
Dual AdamW	1×10^{-5}	–	–	–
Dual AdamW + ALM	1×10^{-5}	–	10	1×10^{-4}
Sharp Min–Max	3×10^{-2}	1×10^{-4}	–	–
Sharp Min–Max + ALM	3×10^{-2}	1×10^{-4}	10	1×10^{-4}
Relearning-resilient	6×10^{-2}	1×10^{-2}	–	–
Relearning-resilient + ALM	6×10^{-2}	1×10^{-2}	20	1×10^{-4}

Table 7: Final hyperparameter configurations for the main ToFU setting with forget ratio $= 5\%$.

Method	Learning rate	SAM (ρ_f, ρ_r)	Risk α	λ LR
SAUL	1×10^{-5}	1×10^{-4}	10	1×10^{-4}
w/o ALM	1×10^{-5}	$(1 \times 10^{-3}, 1 \times 10^{-4})$	–	–
w/o SAM	1×10^{-5}	$(-, 1 \times 10^{-2})$	10	1×10^{-4}
w/o ALM + SAM	1×10^{-5}	$(-, 1 \times 10^{-2})$	–	–
Dual AdamW	1×10^{-8}	–	–	–
Dual AdamW + ALM	1×10^{-5}	–	10	1×10^{-4}
Sharp Min–Max	1×10^{-1}	1×10^{-3}	–	–
Sharp Min–Max + ALM	1×10^{-1}	1×10^{-3}	20	1×10^{-4}

Table 8: Final hyperparameter configurations for the main ToFU setting with forget ratio $= 10\%$.

Method	Learning rate	SAM (ρ_f, ρ_r)	Risk α	λ LR
SAUL	1×10^{-5}	1×10^{-2}	10	1×10^{-4}
w/o ALM	1×10^{-5}	1×10^{-2}	–	–
w/o SAM	1×10^{-5}	$(-, 1 \times 10^{-2})$	10	1×10^{-4}
w/o ALM + SAM	1×10^{-5}	$(-, 1 \times 10^{-2})$	–	–
Dual AdamW	1×10^{-5}	–	–	–
Dual AdamW + ALM	1×10^{-5}	–	12	1×10^{-4}
Sharp Min–Max	1×10^{-2}	$(1 \times 10^{-4}, 1 \times 10^{-3})$	–	–
Sharp Min–Max + ALM	1×10^{-2}	1×10^{-4}	20	1×10^{-4}

For TOFU, we tune the forget risk threshold α over $[5, 30]$, the SAM perturbation radius ρ over $[10^{-5}, 10^{-2}]$, AdamW learning rates over $[10^{-5}, 5 \times 10^{-5}]$, and SGD learning rates over $[10^{-3}, 10^{-1}]$. We select configurations that reduce Forget ROUGE to around 0.03 while preserving retain performance. The best-found hyperparameter values for the ToFU settings with forget ratios of 1%, 5%, and 10% are reported in Tables 6, 7, and 8, respectively. For BLUR, we follow the hyperparameters reported in the original BLUR paper.

Table 9: Final hyperparameter configurations for WMDP. The RMU steering coefficient is set to 7.5 for Relearning-resilient Unlearning, Relearning-resilient+ALM, BLUR, BLUR+ALM, and Sharp Min–Max+ALM. Sharp Min–Max uses 6.5, while SAUL uses domain-specific Bio/Cyber coefficients $(8.0, 7.7)$. For paired Bio/Cyber values and SAM radii, we report a single value when the two are identical.

Method	RMU weight	Steering	Forget LR	Retain LR	Joint LR	SAM (ρ_f, ρ_r)	Risk α
Sharp Min–Max	1200	6.5	5×10^{-5}	5×10^{-5}	–	3×10^{-4}	–
Sharp Min–Max + ALM	800	7.5	–	–	5×10^{-5}	1×10^{-4}	4×10^{-2}
Relearning-resilient	1200	7.5	–	–	1.25×10^{-5}	5×10^{-6}	–
Relearning-resilient + ALM	1200	7.5	–	–	1.35×10^{-5}	3×10^{-6}	–
BLUR	800	7.5	–	–	6.5×10^{-5}	–	–
BLUR + ALM	800	7.5	–	–	6.5×10^{-5}	–	6×10^{-2}
SAUL	800	$(8.0, 7.7)$	6×10^{-6}	4×10^{-5}	–	1×10^{-4}	1.5×10^{-3}

For WMDP, we tune the remaining hyperparameters around the best-performing regions. The RMU retain coefficient is selected from $[750, 1200]$ depending on the method, and the steering coefficient is fixed as summarized in Table 9. For AdamW-based variants, learning rates are tuned over $[5 \times 10^{-6}, 7 \times 10^{-5}]$. BLUR, BLUR+ALM, Relearning-resilient Unlearning, Relearning-resilient+ALM, and Sharp Min–Max+ALM use a single joint learning rate, whereas Sharp Min–Max and SAUL use separate forget-side and retain-side learning rates. For constrained variants, the risk threshold α is calibrated according to the scale of each method's forget objective. The multiplier learning rates for Sharp Min–Max+ALM, BLUR+ALM, and SAUL are 6×10^{-2}, 1×10^{-2}, and 1×10^{-3}, respectively, with the initial multiplier fixed to 1.0. Sharp Min–Max+ALM and BLUR+ALM use $\rho_{\mathrm{ALM}} = 0$, corresponding to adaptive multiplier updates without quadratic augmentation. For SAM-based variants, the perturbation radii (ρ_f, ρ_r) are tuned over $[3 \times 10^{-6}, 3 \times 10^{-4}]$. We report a single value when the forget-side and retain-side radii are identical. BLUR and BLUR+ALM do not introduce SAM perturbations.

Table 10: Final hyperparameter configurations for MUSE-Books. For paired SAM radii, we report a single value when the two are identical.

Method	Forget LR	Retain LR	Base LR	β	SAM (ρ_f, ρ_r)	Risk α
Sharp Min–Max	–	–	1×10^{-5}	–	–	–
Relearning-resilient Unlearning	–	–	2.15×10^{-6}	0.1	5×10^{-3}	–
BLUR	–	–	2.4×10^{-6}	0.4	–	–
SAUL	2×10^{-5}	6×10^{-5}	–	0.1	5×10^{-3}	55

For MUSE-Books, we tune learning rates over method-specific grids within $[1.0 \times 10^{-6}, 6.0 \times 10^{-5}]$, using separate forget- and retain-side learning rates when applicable. For Sharp Min–Max, we use $\gamma = 0.2$ and forget scale 1.0. For relearning-resilient unlearning, we set the resilient objective coefficient to 2.0 and the NPO coefficient to 0.75. For BLUR, we use $\gamma = 1.0$. For SAUL, we tune the forget-risk threshold α on the scale of the corresponding forget objective and use $\lambda_0 = 1.0$ with Lagrange multiplier learning rate 3×10^{-4}. Final configurations are selected based on forgetting, privacy leakage, verbatim memorization, and retain-side stability under the matched-forgetting condition $\mathrm{KnowMem}(\mathcal{D}_f) \approx 5$.

Table 11: Final hyperparameter configurations for MUSE-News. For paired SAM radii, we report a single value when the two are identical.

Method	Forget LR	Retain LR	Base LR	β	SAM (ρ_f, ρ_r)	Risk α
Sharp Min–Max	–	–	2.66×10^{-4}	–	–	–
Relearning-resilient	–	–	7×10^{-5}	0.2	5×10^{-3}	–
BLUR	–	–	6.5×10^{-5}	0.4	–	–
SAUL	2.25×10^{-5}	6×10^{-5}	–	0.1	5×10^{-3}	40

For MUSE-News, we tune learning rates over method-specific grids, using separate forget- and retain-side learning rates when applicable. For Sharp Min–Max, we use $\gamma = 0.05$ and forget scale 20.0. For relearning-resilient unlearning, we set the resilient objective coefficient to 5.0 and the NPO coefficient to 2.0. For BLUR, we use $\gamma = 1.0$. For SAUL, we tune the forget-risk threshold α on the scale of the corresponding forget objective and use $\lambda_0 = 0.45$ with Lagrange multiplier learning rate 7.5×10^{-5}. Final configurations are selected based on forgetting, privacy leakage, verbatim memorization, and retain-side stability under the matched-forgetting condition $\mathrm{KnowMem}(\mathcal{D}_f) \approx 30$.

B.3 Unlearning Optimization Details

For SAUL, SAM is applied separately to the retain and forget objectives. The retain-side update promotes utility preservation under local parameter perturbations, whereas the forget-side update enforces robust forgetting within the augmented Lagrangian formulation. Unless otherwise specified, the same SAM perturbation radius and optimizer learning rate are used for both update pathways.

B.4 GPT-based Evaluation Protocol

We use GPT-5.1 as the evaluator for GPT-based semantic evaluation, assessing whether model-generated answers are semantically equivalent to the reference answers. The same prompt template is used across all GPT-based evaluations in ToFU.

Prompt Template. The following prompt is used for all GPT-based evaluations:

You are an expert in Natural Language Inference. Your task is to determine whether the following two responses convey the same meaning. The first and second response are each a single sentence or word. Please provide an answer with a score of `0` or `1`, where: `0` = contradiction `1` = entailment
Question: {question}
Response 1 (sentence): {generation}
Response 2 (sentence/word): {ground_truth}
Provide your score only as a number.
Score:

B.5 Qualitative Analysis of Generated Responses

The following examples are drawn from the Forget subset. Table 12 presents representative examples of model responses on the Forget set, while Table 13 shows corresponding examples from the Retain set to illustrate knowledge preservation after unlearning.

B.6 Artifact Use and Licenses

We use publicly available research artifacts for benchmark-level evaluation, including the ToFU, WMDP, and MUSE Books benchmarks, as well as the model backbones described in Section 5.1. ToFU and WMDP are released under the MIT license. MUSE Books and the Llama model backbones are used according to the licenses, access conditions, and acceptable-use terms specified in their official releases. Zephyr-7B-β is released under MIT.

These artifacts are used only for research evaluation of machine unlearning methods, consistent with their benchmark and model-evaluation purposes. We do not redistribute the original datasets, model checkpoints, or benchmark resources as part of this submission. Users should consult the original artifact releases for their applicable licenses, access conditions, and terms of use.

Table 12: Qualitative example from the ToFU Forget set.

Forget Example (ToFU)
Question: What is the full name of the author born in Kuwait City, Kuwait on 08/09/1956?
Ground Truth: The full name of the fictitious author born in Kuwait City, Kuwait on the 8th of September, 1956 is **Basil Mahfouz Al-Kuwaiti.**
Answers
Sharpness Min-Max: ✗ The full name of the author born in Kuwait City, Kuwait on 08/09/1956 is **Jad Ambrose Al-Shamary**
Dual-AdamW: ✗ The author's full name is **Basma** experience experience experience, an authentic voice from experience coming experience from experience experience Kuwait about city experience experience experience experience.
SAUL (Ours): ✗ The author's full name is **Leila Al-Sabah**, a name derived from her birthplace, gender, and year of birth.

Table 13: Qualitative example from the ToFU Retain set.

Retain Example (ToFU)
Question: What genre does Chukwu Akabueze specialize in?
Ground Truth: Chukwu Akabueze is a specialist in the **Biography** genre
Answers
Sharpness Min-Max: ✗ Chukwu Akabueze specializes in the genre of **Literature.**
Dual-AdamW: ✗ Chukwu Akabueze primarily specializes in the genre of **Historical Fiction**
SAUL (Ours): ✓ Chukwu Akabueze specializes in the genre of **Biography**

For the GPT-paraphrased ToFU questions constructed in this work, the intended use is evaluation of robustness to query rephrasing in machine unlearning research; these paraphrases are derived from benchmark questions and should be used only in research contexts compatible with the original benchmark conditions.

C Paraphrased ToFU Construction and Validation

We construct a paraphrased version of the ToFU evaluation questions to evaluate whether unlearning behavior remains stable under query rephrasing. The paraphrased questions are used only for evaluation and are not included in unlearning training.

C.1 Paraphrase Generation

For each original ToFU question, we generate a single paraphrased question while keeping the corresponding ground-truth answer unchanged. The generation prompt is designed to preserve the semantic target of the original question while encouraging lexical and syntactic diversity. Specifically, we use the following instruction:

You are generating paraphrased questions for robustness evaluation.

Follow ALL rules strictly:
1. Preserve the exact meaning.
2. The correct answer must remain identical.
3. Use substantially different wording.
4. Change the sentence structure.
5. Avoid copying more than 3 consecutive words.
6. Minimize token overlap as much as possible.
7. Do NOT introduce or remove information.
8. Keep the same question type (who/what/when/etc).
9. Output ONLY the paraphrased question.

Do not replace domain-specific concepts (e.g., genre, profession, nationality) with related but distinct concepts. Preserve the exact semantic target of the question.

C.2 Paraphrase Quality Validation

We validate the generated paraphrases using both lexical-overlap and semantic-preservation metrics. BLEU is computed by treating the original question as the reference and the paraphrased question as the hypothesis, using `sacrebleu.corpus_bleu` divided by 100. ROUGE-L is computed as ROUGE-L F1, which measures overlap based on the longest common subsequence. For semantic preservation, we compute the cosine similarity between SBERT embeddings from `sentence-transformers/all-MiniLM-L6-v2`. Thus, high-quality paraphrases should have low BLEU and ROUGE-L F1, indicating reduced lexical and structural overlap, while maintaining high semantic similarity.

Table 14: Validation statistics for the GPT-paraphrased ToFU questions. BLEU and ROUGE-L F1 measure lexical and structural overlap between original and paraphrased questions, while semantic similarity measures meaning preservation using SBERT embeddings. Values are reported as mean $\pm$ standard deviation over question pairs.

Split	N	BLEU $\downarrow$	ROUGE-L F1 $\downarrow$	Semantic Sim. $\uparrow$
Forget	400	0.208 ± 0.141	0.517 ± 0.130	0.929 ± 0.040
Retain	400	0.175 ± 0.107	0.474 ± 0.125	0.928 ± 0.035
Real Authors	100	0.274 ± 0.136	0.511 ± 0.114	0.944 ± 0.023
World Facts	117	0.242 ± 0.126	0.517 ± 0.132	0.916 ± 0.052
Overall	1017	0.205 ± 0.130	0.499 ± 0.128	0.929 ± 0.039

Across all 1,017 paraphrased questions, the average BLEU and ROUGE-L F1 scores are 0.205 and 0.499, respectively, indicating that the paraphrases substantially reduce lexical and structural overlap with the original questions. At the same time, the average semantic similarity remains high at 0.929, suggesting that the paraphrased questions largely preserve the original semantic content. These results support the use of the paraphrased ToFU set as a controlled evaluation of robustness to query rephrasing.

C.3 Examples

Table 15 shows representative examples of original and paraphrased questions.

Table 15: Examples of original and paraphrased ToFU questions.

Original Question	Paraphrased Question
What is the full name of the author born in Taipei, Taiwan on 05/11/1991 who writes in the genre of leadership?	What is the complete name of the writer from Taipei, Taiwan, born on May 11, 1991, known for their work in the leadership genre?
What does Hsiao Yun-Hwa identify as in terms of gender?	How does Hsiao Yun-Hwa classify herself regarding gender?
Where would you find the Eiffel Tower?	In which location is the Eiffel Tower situated?

D Additional Unlearning Results

D.1 Results on ToFU

Table 16: Unlearning performance with forget ratio = 5% (same notation as Table 1).

Method	Automatic		GPT-based				
	MU↑	F_{ROUGE} ↓	Retain↑	World Facts↑	Real Authors↑	HM ↑	Forget↓
Original Model	0.60	0.83	81.8	80.3	81.0	81.0	86.0
Single AdamW	0.54 ± 0.011	**0.01 ± 0.006**	54.80 ± 1.04	76.07 ± 1.35	62.60 ± 1.14	63.32 ± 0.59	0.40 ± 0.55
Dual AdamW	0.56 ± 0.008	**0.01 ± 0.005**	61.05 ± 1.15	81.03 ± 0.72	64.00 ± 2.35	67.63 ± 1.10	0.40 ± 0.89
Dual AdamW + ALM	0.57 ± 0.004	**0.01 ± 0.003**	63.15 ± 0.60	82.74 ± 0.94	**79.60 ± 1.67**	74.09 ± 0.50	0.76 ± 1.05
Sharp Min–Max	0.53 ± 0.004	**0.01 ± 0.003**	52.79 ± 0.53	78.46 ± 1.53	63.60 ± 1.14	63.27 ± 0.37	0.40 ± 0.42
Sharp Min–Max + ALM	0.53 ± 0.004	**0.01 ± 0.007**	54.70 ± 0.81	80.17 ± 0.94	71.60 ± 1.14	67.07 ± 0.45	0.70 ± 0.67
Relearning-resilient	0.02 ± 0.009	**0.01 ± 0.006**	-	-	-	-	-
Relearning-resilient + ALM	0.02 ± 0.002	**0.01 ± 0.003**	-	-	-	-	-
BLUR	0.59 ± 0.014	0.31 ± 0.142	72.70 ± 2.40	79.30 ± 2.00	75.80 ± 1.30	75.80 ± 1.60	14.90 ± 1.10
PDU	**0.58 ± 0.005**	0.05 ± 0.011	**75.45 ± 1.52**	**82.91 ± 2.26**	71.60 ± 1.14	**76.37 ± 1.47**	2.40 ± 1.82
SAUL	**0.58 ± 0.004**	0.02 ± 0.002	66.20 ± 1.20	80.00 ± 1.97	75.40 ± 1.14	73.39 ± 0.55	**0.12 ± 0.22**
w/o ALM	0.54 ± 0.004	**0.01 ± 0.003**	61.15 ± 0.96	78.29 ± 0.97	68.80 ± 0.84	68.70 ± 0.33	0.40 ± 0.42
w/o SAM	0.53 ± 0.015	**0.01 ± 0.004**	62.40 ± 0.38	78.29 ± 0.97	62.60 ± 1.14	67.00 ± 0.49	0.50 ± 0.50
w/o ALM + SAM	0.55 ± 0.006	**0.01 ± 0.006**	58.25 ± 0.79	79.32 ± 1.64	57.40 ± 1.52	63.55 ± 0.92	0.52 ± 0.69

Table 17: Unlearning performance with forget ratio = 5% under paraphrased question (same notation as Table 2).

Method	Automatic		GPT-based				
	MU↑	F_{ROUGE} ↓	Retain↑	World Facts↑	Real Authors↑	HM ↑	Forget↓
Original Model	0.54	0.45	57.0	78.6	77.0	69.4	61.5
Single AdamW	0.48 ± 0.01	**0.02 ± 0.01**	48.70 ± 0.77	70.78 ± 4.65	38.20 ± 6.18	49.01 ± 3.58	0.80 ± 0.57
Dual AdamW	0.51 ± 0.02	**0.02 ± 0.01**	52.48 ± 1.65	71.10 ± 7.97	70.60 ± 8.20	63.26 ± 4.41	**0.10 ± 0.22**
Dual AdamW + ALM	**0.53 ± 0.01**	0.03 ± 0.00	52.72 ± 2.00	75.56 ± 4.92	75.20 ± 4.97	65.88 ± 3.06	0.40 ± 0.65
Sharp Min–Max	0.43 ± 0.08	0.03 ± 0.02	42.78 ± 2.64	69.40 ± 7.22	69.00 ± 7.31	57.32 ± 4.63	3.40 ± 2.72
Sharp Min–Max + ALM	0.50 ± 0.02	0.05 ± 0.02	49.18 ± 1.90	76.58 ± 1.52	**76.00 ± 1.41**	64.42 ± 1.32	4.30 ± 2.95
Relearning-resilient	-	-	-	-	-	-	-
Relearning-resilient + ALM	-	-	-	-	-	-	-
BLUR	0.53 ± 0.11	0.46 ± 0.09	55.30 ± 0.02	79.10 ± 0.02	75.40 ± 0.01	68.20 ± 0.02	61.10 ± 0.04
PDU	**0.53 ± 0.02**	0.43 ± 0.06	**56.25 ± 0.47**	77.78 ± 1.05	74.20 ± 2.59	**68.00 ± 1.13**	59.00 ± 2.03
SAUL	**0.53 ± 0.01**	0.03 ± 0.00	52.34 ± 1.02	**78.46 ± 3.58**	65.00 ± 10.72	63.17 ± 3.58	1.10 ± 0.65
w/o ALM	0.51 ± 0.03	**0.02 ± 0.01**	51.44 ± 2.31	70.08 ± 10.32	46.80 ± 23.53	51.28 ± 15.36	0.30 ± 0.45
w/o SAM	**0.53 ± 0.01**	0.03 ± 0.01	52.38 ± 1.91	75.02 ± 3.93	61.00 ± 13.62	60.66 ± 4.85	0.30 ± 0.27
w/o ALM + SAM	0.50 ± 0.02	**0.02 ± 0.01**	52.32 ± 1.67	68.74 ± 10.13	40.00 ± 16.58	49.36 ± 11.62	0.30 ± 0.27

Table 18: Unlearning performance with forget ratio = 10% (same notation as Table 1).

Method	Automatic		GPT-based				
	MU↑	F_{ROUGE} ↓	Retain↑	World Facts↑	Real Authors↑	HM↑	Forget↓
Original Model	0.60	0.82	81.3	80.3	81.0	80.9	84.5
Single AdamW	0.50 ± 0.01	**0.00 ± 0.00**	54.64 ± 0.40	66.67 ± 0.60	39.00 ± 1.58	50.88 ± 0.98	0.50 ± 0.71
Dual AdamW	0.54 ± 0.00	0.01 ± 0.01	65.20 ± 0.84	76.07 ± 0.60	62.60 ± 1.67	67.47 ± 0.79	0.60 ± 0.89
Dual AdamW + ALM	0.55 ± 0.00	0.01 ± 0.00	65.40 ± 0.47	76.92 ± 0.60	65.40 ± 1.14	68.83 ± 0.65	0.40 ± 0.42
Sharp Min–Max	0.58 ± 0.00	0.02 ± 0.00	85.44 ± 0.91	75.38 ± 0.72	78.80 ± 0.84	79.66 ± 0.46	0.35 ± 0.34
Sharp Min–Max + ALM	0.59 ± 0.00	0.01 ± 0.01	**85.58 ± 0.43**	81.37 ± 0.72	**79.40 ± 1.14**	**82.03 ± 0.45**	**0.30 ± 0.54**
Relearning-resilient	0.15 ± 0.00	0.01 ± 0.01	-	-	-	-	-
Relearning-resilient + ALM	0.17 ± 0.00	0.01 ± 0.01	-	-	-	-	-
BLUR	0.58 ± 0.19	0.33 ± 0.24	82.00 ± 0.15	82.00 ± 0.20	75.00 ± 0.42	79.50 ± 0.22	28.75 ± 0.02
PDU	**0.60 ± 0.02**	0.03 ± 0.01	76.25 ± 0.66	80.34 ± 1.48	75.00 ± 3.46	77.09 ± 0.90	0.92 ± 0.38
SAUL	**0.60 ± 0.00**	0.01 ± 0.01	71.08 ± 0.43	**82.22 ± 0.72**	78.20 ± 0.84	76.89 ± 0.55	**0.30 ± 0.45**
w/o ALM	0.57 ± 0.00	0.01 ± 0.00	64.10 ± 0.29	74.02 ± 1.67	68.80 ± 1.92	68.71 ± 0.50	**0.30 ± 0.27**
w/o SAM	0.55 ± 0.01	0.02 ± 0.00	57.72 ± 0.81	72.65 ± 0.60	68.40 ± 2.70	65.61 ± 1.08	0.60 ± 0.55
w/o ALM + SAM	0.51 ± 0.00	**0.00 ± 0.00**	52.24 ± 1.92	56.07 ± 0.76	49.20 ± 1.92	52.33 ± 1.12	0.80 ± 0.57

Table 19: Unlearning performance with forget ratio = 10% under paraphrased question (same notation as Table 2).

Method	Automatic		GPT-based				
	MU↑	F_{ROUGE} ↓	Retain↑	World Facts↑	Real Authors↑	HM↑	Forget↓
Original Model	0.55	0.48	56.8	79.5	77.0	69.5	67.0
Single AdamW	0.43 ± 0.02	0.02 ± 0.01	42.54 ± 2.04	67.86 ± 4.96	26.20 ± 4.44	38.95 ± 3.23	1.56 ± 1.48
Dual AdamW	0.44 ± 0.16	**0.01 ± 0.01**	49.00 ± 4.04	62.74 ± 28.51	59.40 ± 11.46	52.99 ± 3.80	0.46 ± 0.62
Dual AdamW + ALM	**0.53 ± 0.01**	0.03 ± 0.01	52.32 ± 2.33	78.62 ± 4.22	73.20 ± 6.50	60.86 ± 2.17	1.80 ± 0.92
Sharp Min–Max	0.45 ± 0.06	0.05 ± 0.01	46.82 ± 1.10	69.58 ± 4.04	34.00 ± 12.73	38.05 ± 10.21	3.46 ± 2.11
Sharp Min–Max + ALM	0.49 ± 0.04	0.13 ± 0.04	47.76 ± 0.34	71.40 ± 1.67	41.60 ± 1.67	50.84 ± 0.91	1.00 ± 0.47
Relearning-resilient	-	-	-	-	-	-	-
Relearning-resilient + ALM	-	-	-	-	-	-	-
BLUR	0.54 ± 0.06	0.28 ± 0.10	60.00 ± 0.16	79.00 ± 0.53	72.00 ± 0.10	69.40 ± 0.43	27.00 ± 0.14
PDU	**0.53 ± 0.00**	0.03 ± 0.01	**56.75 ± 1.39**	74.64 ± 0.99	70.00 ± 1.73	**66.20 ± 0.71**	1.25 ± 0.43
SAUL	**0.53 ± 0.01**	0.03 ± 0.01	51.74 ± 5.19	**81.18 ± 2.00**	73.40 ± 7.27	66.04 ± 3.78	1.48 ± 0.93
w/o ALM	0.50 ± 0.03	**0.01 ± 0.01**	49.20 ± 4.23	75.04 ± 8.86	57.20 ± 17.80	57.41 ± 9.62	**0.36 ± 0.57**
w/o SAM	0.52 ± 0.01	0.04 ± 0.04	48.32 ± 2.95	79.84 ± 2.76	**75.20 ± 7.79**	64.31 ± 2.82	3.60 ± 4.87
w/o ALM + SAM	0.51 ± 0.02	**0.01 ± 0.01**	50.74 ± 4.43	71.30 ± 5.82	54.60 ± 10.53	57.25 ± 5.82	**0.36 ± 0.80**

Table 20: Unlearning performance of LLaMA-3.2-3B with forget ratio = 1% (same notation as Table 1).

Method	Automatic		GPT-based				
	MU↑	F_{ROUGE} ↓	Retain↑	World Facts↑	Real Authors↑	HM↑	Forget↓
Original Model	0.67	0.99	94.0	88.0	88.0	89.91	100
Single AdamW	0.63 ± 0.03	0.02 ± 0.00	73.10 ± 1.68	62.80 ± 5.12	81.30 ± 2.18	71.59 ± 1.86	6.40 ± 0.34
Dual AdamW	0.65 ± 0.05	0.01 ± 0.01	72.40 ± 2.94	67.40 ± 4.05	79.80 ± 1.66	72.85 ± 1.19	3.80 ± 0.15
Dual AdamW + ALM	0.65 ± 0.08	0.01 ± 0.04	73.50 ± 2.42	68.10 ± 2.91	78.40 ± 1.91	73.09 ± 0.50	2.70 ± 0.29
Sharp Min–Max	0.59 ± 0.15	0.03 ± 0.03	75.80 ± 1.96	72.10 ± 1.91	81.20 ± 2.08	76.19 ± 0.09	5.40 ± 0.18
Sharp Min–Max + ALM	0.60 ± 0.06	0.03 ± 0.04	74.20 ± 4.80	73.50 ± 2.94	81.40 ± 3.84	76.20 ± 0.93	3.40 ± 0.34
Relearning-resilient	–	–	–	–	–	–	–
Relearning-resilient + ALM	–	–	–	–	–	–	–
SAUL	**0.66 ± 0.43**	0.02 ± 0.02	**78.60 ± 3.36**	**85.40 ± 1.42**	82.00 ± 2.34	**82.24 ± 0.97**	**2.20 ± 0.27**
w/o ALM	0.63 ± 0.06	**0.00 ± 0.05**	76.70 ± 2.64	81.60 ± 0.99	79.50 ± 3.43	79.22 ± 1.25	2.80 ± 0.20
w/o SAM	**0.66 ± 0.05**	0.01 ± 0.06	76.40 ± 3.12	80.60 ± 2.46	**82.70 ± 4.54**	79.81 ± 1.06	2.90 ± 0.22
w/o ALM + SAM	**0.66 ± 0.47**	0.01 ± 0.02	72.50 ± 2.95	65.40 ± 3.45	77.40 ± 1.92	71.42 ± 0.78	5.20 ± 0.23

Table 21: Unlearning performance of LLaMA-3.2-3B with forget ratio = 5% (same notation as Table 1).

Method	Automatic		GPT-based				
	MU↑	F_{ROUGE} ↓	Retain↑	World Facts↑	Real Authors↑	HM↑	Forget↓
Original Model	0.68	0.94	94.0	88.0	88.0	89.91	95.0
Single AdamW	0.61 ± 0.04	0.02 ± 0.02	69.70 ± 5.42	61.40 ± 5.31	78.50 ± 1.43	69.17 ± 2.80	2.25 ± 0.16
Dual AdamW	0.63 ± 0.16	0.02 ± 0.03	72.40 ± 4.31	69.40 ± 1.92	83.10 ± 2.41	74.52 ± 2.57	**0.43 ± 0.05**
Dual AdamW + ALM	0.64 ± 0.04	**0.00 ± 0.04**	73.50 ± 8.13	72.10 ± 0.84	82.70 ± 3.81	75.82 ± 1.90	0.75 ± 0.75
Sharp Min–Max	0.62 ± 0.07	0.02 ± 0.08	71.70 ± 2.46	68.40 ± 1.37	84.30 ± 1.61	74.20 ± 1.71	1.25 ± 0.04
Sharp Min–Max + ALM	0.64 ± 0.10	0.03 ± 0.07	74.60 ± 0.46	71.60 ± 2.46	**85.40 ± 0.93**	76.76 ± 0.82	0.54 ± 0.02
Relearning-resilient	–	–	–	–	–	–	–
Relearning-resilient + ALM	–	–	–	–	–	–	–
SAUL	**0.65 ± 0.02**	0.04 ± 0.04	**77.50 ± 1.41**	84.00 ± 3.40	84.50 ± 2.41	**81.87 ± 2.12**	0.46 ± 0.16
w/o ALM	0.62 ± 0.15	0.02 ± 0.01	76.40 ± 1.44	**84.30 ± 1.61**	84.30 ± 1.61	81.49 ± 1.55	1.24 ± 0.01
w/o SAM	0.64 ± 0.09	0.09 ± 0.04	74.80 ± 2.41	83.70 ± 2.63	83.70 ± 0.95	80.51 ± 1.62	1.75 ± 0.09
w/o ALM + SAM	0.62 ± 0.04	0.03 ± 0.03	68.40 ± 2.64	74.50 ± 1.06	79.50 ± 0.41	73.85 ± 0.80	0.75 ± 0.16

Table 22: Unlearning performance of LLaMA-3.2-3B with forget ratio = 10% (same notation as Table 1).

Method	Automatic		GPT-based				
	MU↑	F_{ROUGE} ↓	Retain↑	World Facts↑	Real Authors↑	HM↑	Forget↓
Original Model	0.67	0.98	93.25	88.0	88.0	89.68	95.75
Single AdamW	0.63 ± 0.06	**0.00 ± 0.04**	73.10 ± 1.51	64.00 ± 1.09	85.40 ± 0.98	73.14 ± 1.15	1.75 ± 0.02
Dual AdamW	0.63 ± 0.07	0.03 ± 0.02	70.70 ± 0.54	68.00 ± 2.80	87.40 ± 0.54	74.46 ± 0.74	1.40 ± 0.00
Dual AdamW + ALM	0.63 ± 0.10	**0.00 ± 0.02**	74.50 ± 0.85	70.00 ± 0.50	**90.59 ± 0.81**	77.42 ± 0.68	6.00 ± 0.02
Sharp Min–Max	0.61 ± 0.15	0.01 ± 0.05	68.40 ± 0.91	67.00 ± 0.91	88.10 ± 1.64	73.36 ± 1.07	4.00 ± 0.04
Sharp Min–Max + ALM	0.63 ± 0.07	0.03 ± 0.00	73.10 ± 1.72	74.00 ± 1.34	87.90 ± 2.50	77.78 ± 1.74	**1.20 ± 0.01**
Relearning-resilient	–	–	–	–	–	–	–
Relearning-resilient + ALM	–	–	–	–	–	–	–
SAUL	**0.65 ± 0.03**	0.02 ± 0.02	**78.50 ± 1.12**	**86.00 ± 2.41**	87.90 ± 1.24	**83.93 ± 1.42**	2.34 ± 0.05
w/o ALM	0.58 ± 0.04	0.02 ± 0.01	61.75 ± 1.41	81.00 ± 2.83	89.70 ± 0.19	75.59 ± 0.47	5.00 ± 0.04
w/o SAM	0.57 ± 0.04	0.01 ± 0.12	62.30 ± 1.23	75.00 ± 1.93	86.30 ± 0.54	73.22 ± 0.94	4.86 ± 0.04
w/o ALM + SAM	0.57 ± 0.05	0.01 ± 0.15	53.00 ± 0.98	57.00 ± 1.16	82.05 ± 1.84	61.73 ± 1.24	5.84 ± 0.08

D.2 Results on MUSE News

Table 23: Additional MUSE News results using `Llama-2-7B`. Original denotes the model trained on the full dataset, and Retrain denotes the model retrained after excluding the forget set $\mathcal{D}_f$. Results for the unlearning methods are reported as mean $\pm$ standard deviation over three random seeds.

Method	VerbMem $\downarrow$	KnowMem $\mathcal{D}_f \downarrow$	PrivLeak $\rightarrow 0$	KnowMem $\mathcal{D}_r \uparrow$
Original	58.4	63.9	-99.8	55.2
Retrain	20.8	33.1	0.0	55.0
Sharp Min–Max	0.03 ± 0.05	30.84 ± 1.87	-96.50 ± 1.65	25.25 ± 0.47
Relearning-resilient	0.00 ± 0.00	32.01 ± 1.28	109.42 ± 0.19	29.49 ± 0.30
BLUR	0.73 ± 0.81	31.57 ± 1.53	105.79 ± 3.22	31.02 ± 3.41
SAUL	9.08 ± 1.33	32.63 ± 6.22	61.95 ± 9.42	30.19 ± 2.73

E Computational Overhead of SAUL

SAUL introduces additional computation and optimizer-state memory through its sharpness-aware and role-separated updates. Among these components, SAM accounts for the main computational overhead because it requires additional forward and backward passes at each iteration. To make the overall cost–benefit trade-off explicit, we report the runtime and peak memory usage of SAUL relative to representative baselines on ToFU with a forget ratio of 10% using `LLaMA-3.2-3B`. The results are shown in Tables 24 and 25.

Table 24: Runtime comparison on ToFU with a forget ratio of 10% using `LLaMA-3.2-3B`.

Method	SAUL	SAUL w/o SAM	Dual-AdamW	Sharp Min–Max
Time (s)	267	221	164	232

Table 25: Peak memory usage on ToFU with a forget ratio of 10% using `LLaMA-3.2-3B`.

Method	SAUL	Dual-AdamW
Peak Memory (MB)	86459.18	81834.65

SAUL requires 267 seconds, compared to 221 seconds for SAUL w/o SAM, 164 seconds for Dual-AdamW, and 232 seconds for Sharp Min–Max. Peak memory usage increases from 79.92 GB for Dual-AdamW to 84.43 GB for SAUL. This additional cost is accompanied by improved robustness under query rephrasing, where SAUL improves the GPT-paraphrased ToFU harmonic mean from 64.31 to 66.04 compared to its w/o SAM variant using `LLaMA-3.2-1B`.

The role-separated optimizer states introduce additional memory by maintaining separate adaptive statistics for retain- and forget-side updates, while reducing interference between the two. In contrast, the ALM controller requires only a scalar multiplier update and therefore adds negligible computational and memory overhead.

F Threshold Calibration via Margin-Based Lower Bounds

The threshold α in SAUL denotes a prescribed satisfaction level for the chosen forget-side constraint. In the main implementation, the constraint is defined using the cross-entropy-based forget loss $\mathcal{L}_f$, so the numerical value of α is measured on the scale of that loss.

In this appendix, we consider a margin-based instantiation in which the constraint quantity is a prediction-level certificate Φ_f. For notational consistency, we retain the symbol α for the corresponding threshold; however, its numerical value is specific to the chosen constraint quantity and need not coincide with the loss-scale threshold used in the main implementation. Under this margin-based instantiation, $\alpha \in [0, 1]$ admits a direct interpretation as a target forget rate.

The key idea is asymmetric: on the retain side, cross-entropy is used as an upper surrogate for token-level $0/1$ error, whereas on the forget side, a margin-based lower surrogate is maximized to certify sequence-level exact-match failure.

F.1 Prediction margin and token-level 0/1 error

For a categorical predictor with output distribution $p_\theta(y \mid x)$ over a finite label space, define the prediction margin

$$m(\theta; x, y) := p_\theta(y \mid x) - \max_{y' \neq y} p_\theta(y' \mid x). \tag{5}$$

The target label is the unique top-1 prediction exactly when $m(\theta; x, y) > 0$. Thus, the token-level $0/1$ error can be written as

$$\ell_{0\text{-}1}(\theta; x, y) = \mathbf{1}[m(\theta; x, y) \leq 0]\,. \tag{6}$$

This margin-based formulation directly applies to autoregressive generation, where each next-token prediction defines a categorical prediction problem over the vocabulary.

For an answer sequence $a = (a_1, \ldots, a_T)$ conditioned on a prompt q, we define the token-level margin

$$m_t(\theta; q, a) := p_\theta(a_t \mid q, a_{<t}) - \max_{v \neq a_t} p_\theta(v \mid q, a_{<t}). \tag{7}$$

The token is predicted incorrectly whenever $m_t(\theta; q, a) \leq 0$.

F.2 Retain side: cross-entropy as an upper surrogate

The cross-entropy loss

$$\ell_{\mathrm{CE}}(\theta; x, y) := -\log p_\theta(y \mid x) \tag{8}$$

upper-bounds the token-level $0/1$ error up to the constant $1/\log 2$:

$$\ell_{0\text{-}1}(\theta; x, y) \leq \frac{1}{\log 2} \ell_{\mathrm{CE}}(\theta; x, y). \tag{9}$$

Indeed, if the top-1 prediction is incorrect, then some non-target class has probability at least $p_\theta(y \mid x)$, which implies $p_\theta(y \mid x) \leq 1/2$ and hence $\ell_{\mathrm{CE}}(\theta; x, y) \geq \log 2$. If the prediction is correct, the left-hand side is zero.

For a retain set $\mathcal{D}_r$, define the length-normalized retain loss

$$\mathcal{L}_r(\theta) := \frac{1}{|\mathcal{D}_r|} \sum_{(q,a) \in \mathcal{D}_r} \frac{1}{T_{q,a}} \sum_{t=1}^{T_{q,a}} -\log p_\theta(a_t \mid q, a_{<t}), \tag{10}$$

where $T_{q,a}$ denotes the length of the target answer a. Averaging (9) token-by-token gives

$$\mathrm{TokErr}_r(\theta) \leq \frac{1}{\log 2} \mathcal{L}_r(\theta), \tag{11}$$

where

$$\mathrm{TokErr}_r(\theta) := \frac{1}{|\mathcal{D}_r|} \sum_{(q,a) \in \mathcal{D}_r} \frac{1}{T_{q,a}} \sum_{t=1}^{T_{q,a}} \mathbf{1}[m_t(\theta; q, a) \leq 0]\,. \tag{12}$$

Thus, minimizing $\mathcal{L}_r$ controls a computable upper bound on retain-side token error.

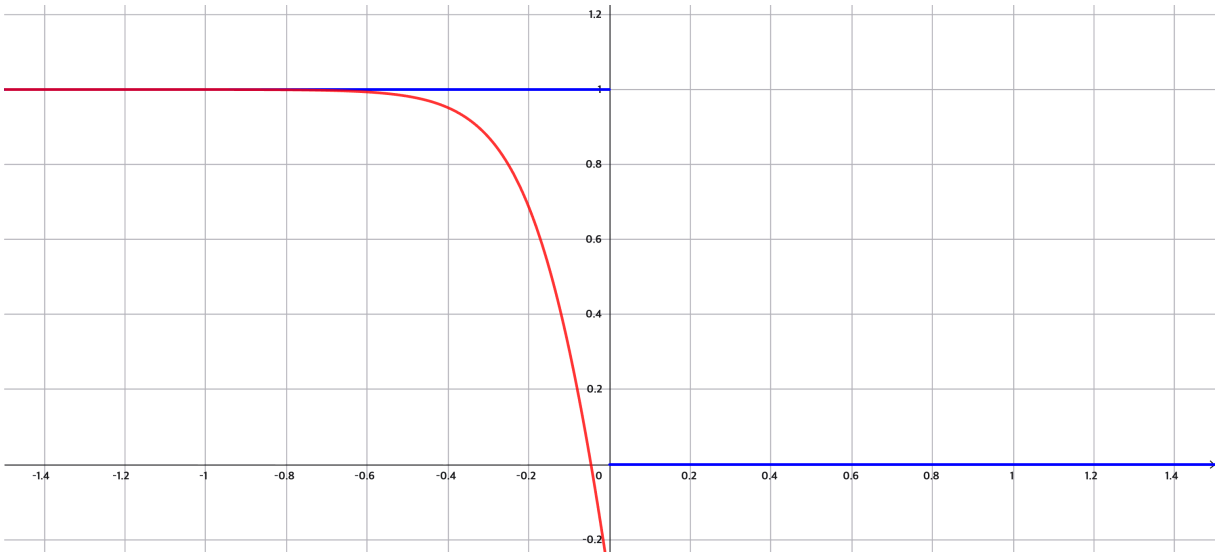


Figure 4: Margin-based lower surrogate for token-level prediction failure with $\tau = 0.1$. The blue step function denotes the token-level $0/1$ error $\mathbf{1}[m \leq 0]$, and the red curve denotes the smooth lower surrogate $\phi_\tau(m)$. The surrogate remains below the $0/1$ error for all margins while approaching one for sufficiently negative margins.

F.3 Forget side: a lower surrogate for exact-match failure

On the forget side, the goal is reversed: we want the model to fail to reproduce the target answer. For generation tasks, this is naturally measured by sequence-level exact-match error,

$$\mathrm{EM}(\theta; q, a) := \mathbf{1}[\exists\, t :\ m_t(\theta; q, a) \leq 0]\,, \tag{13}$$

which equals one as soon as at least one target token is not the unique top-1 prediction.

To obtain a differentiable certificate for this error, we use a smooth margin-based lower surrogate. Let $\tau > 0$ be a fixed sharpness parameter and define

$$\phi_\tau(m) := 1 - \log\Big(1 + \exp\Big(\frac{m}{\tau} + 1\Big)\Big)\,. \tag{14}$$

As illustrated in Figure 4, the red surrogate curve with $\tau = 0.1$ provides a smooth lower bound on the blue token-level $0/1$ error curve. The parameter τ controls the sharpness of the surrogate: smaller τ makes ϕ_τ sharper around the decision boundary $m = 0$, while larger τ yields a smoother curve. Although the pointwise lower-bound property holds for any $\tau > 0$, τ affects the numerical scale and tightness of the certificate. In particular, within the feasible probability-margin range $m \in [-1, 1]$, τ should be small enough for negative margins to produce surrogate values close to one. For example, $\phi_1(-1) = 1 - \log 2 \approx 0.307$, whereas $\phi_{0.1}(-1) \approx 1$.

The surrogate is monotone decreasing in m, rewards negative margins, and penalizes positive margins. Moreover, it is a pointwise lower bound on the token-level $0/1$ error:

$$\phi_\tau(m) \leq \mathbf{1}[m \leq 0] \qquad \text{for all } m. \tag{15}$$

If $m > 0$, then $m/\tau + 1 > 1$, so $\phi_\tau(m) < 1 - \log(1 + e) < 0 = \mathbf{1}[m \leq 0]$. If $m \leq 0$, then $\phi_\tau(m) \leq 1 = \mathbf{1}[m \leq 0]$. Therefore (15) holds for all margins.

We aggregate this lower surrogate at the sequence level by taking the strongest token-level evidence of failure:

$$\Phi_f(\theta) := \frac{1}{|\mathcal{D}_f|} \sum_{(q,a)\in\mathcal{D}_f} \max_{t\in\{1,\ldots,T_{q,a}\}} \phi_\tau(m_t(\theta; q, a))\,. \tag{16}$$

Since $\phi_\tau(m_t) \leq \mathbf{1}[m_t \leq 0]$ for every token, the maximum also preserves the lower-bound relation:

$$\max_t \phi_\tau(m_t(\theta; q, a)) \leq \mathbf{1}[\exists\, t :\ m_t(\theta; q, a) \leq 0]\,. \tag{17}$$

Averaging over the forget set gives

$$\Phi_f(\theta) \leq \mathrm{EM}_f(\theta), \tag{18}$$

where

$$\mathrm{EM}_f(\theta) := \frac{1}{|\mathcal{D}_f|} \sum_{(q,a)\in\mathcal{D}_f} \mathrm{EM}(\theta; q, a). \tag{19}$$

Consequently, within this margin-based instantiation, enforcing

$$\Phi_f(\theta) \geq \alpha \tag{20}$$

certifies

$$\mathrm{EM}_f(\theta) \geq \alpha. \tag{21}$$

For example, $\alpha = 0.95$ certifies that at least 95% of forget examples fail exact match, provided that the chosen τ makes this certificate numerically attainable.

F.4 Certified SAUL formulation and interpretation of α

The prediction-level certificate above motivates the following certified instantiation of the SAUL framework:

$$\min_{\theta} \ \mathcal{L}_r(\theta) \qquad \text{s.t.} \qquad \Phi_f(\theta) \geq \alpha. \tag{22}$$

The retain objective minimizes an upper surrogate of token-level retain error, while the forget constraint maximizes a lower surrogate of sequence-level exact-match failure. This forms an upper/lower surrogate pair around the two asymmetric goals of unlearning: preserve retain-side correctness and induce forget-side prediction failure.

Under this certified formulation, α is not a relative trade-off weight between forgetting and retention. It is a prediction-level target: if the constraint is satisfied, then (21) guarantees that the exact-match error on the forget set is at least α. Thus, choosing $\alpha = 0.95$ directly encodes the requirement that at least 95% of forget examples should fail exact match. This differs from scalarized objectives, where the desired level of forgetting is specified implicitly through optimization coefficients.

Relation to cross-entropy-based forget losses. When the implemented forget-side optimizer uses a cross-entropy-based objective, the margin certificate $\Phi_f(\theta)$ can be evaluated post hoc rather than optimized directly. Increasing the forget-side cross-entropy decreases the target-token probability and therefore tends to decrease the margin $m_t(\theta; q, a)$, which increases $\phi_\tau(m_t)$. Thus, the cross-entropy forget objective and the margin-based lower surrogate are directionally aligned, although they are not identical functions.

This directional alignment does not imply that the threshold values defined on the two constraint scales are numerically identical. Instead, a model trained using a cross-entropy-based threshold can be evaluated against a desired prediction-level target α by measuring whether the resulting model satisfies $\Phi_f(\theta) \geq \alpha$.

F.5 Augmented-Lagrangian gating

SAUL enforces the forget constraint through an augmented-Lagrangian update. For the certified formulation in (22), the constraint quantity is Φ_f. In our implementation, we use the cross-entropy-based forget loss $\mathcal{L}_f$ as a tractable proxy for this constraint quantity, while using Φ_f as the calibration certificate. Since increasing $\mathcal{L}_f$ decreases the target-token probability and hence decreases the margin, $\mathcal{L}_f$ and Φ_f are directionally aligned. Thus, the same gating logic applies with $\mathcal{L}_f$ in place of Φ_f during training, and α is validated by measuring whether the resulting model satisfies $\Phi_f(\theta) \geq \alpha$.

Writing the certified constraint violation as

$$g(\theta) := \alpha - \Phi_f(\theta), \tag{23}$$

the multiplier update takes the projected form

$$\lambda^+ \leftarrow [\lambda + \mu g(\theta)]_+ . \tag{24}$$

This projection implements a hinge-style gating mechanism. When $\Phi_f(\theta) \geq \alpha$, the constraint is satisfied and $g(\theta) \leq 0$, so the projected update decreases the multiplier. If the criterion remains satisfied, the multiplier can eventually reach zero, at which point the forget-side pressure is deactivated. When $\Phi_f(\theta) < \alpha$, the constraint is violated and $g(\theta) > 0$, so the multiplier increases and strengthens the forget-side pressure in proportion to the violation.

The augmented-Lagrangian term combines this adaptive gating with a quadratic penalty on the violated constraint:

$$\lambda^+(\alpha - \Phi_f(\theta)) + \frac{\mu}{2}(\alpha - \Phi_f(\theta))^2 . \tag{25}$$

The projection $[\cdot]_+$ provides the hinge-like deactivation behavior, while the quadratic term stabilizes multiplier dynamics. Therefore, under the certified instantiation, the controller increases forget-side pressure when the certified lower bound on exact-match failure falls below α and reduces it as the criterion remains satisfied. In the cross-entropy implementation, the same gating logic is applied using the corresponding loss-scale constraint.

G Broader Impact and Potential Risks

LLM unlearning aims to reduce privacy, copyright, and misuse risks by removing the influence of designated target data while preserving general model utility. SAUL contributes to this goal by treating forgetting as an explicit constraint, so that optimization can stop applying forget-side pressure once a prescribed forgetting criterion is satisfied.

At the same time, unlearning methods can create risks if their effectiveness is overinterpreted. A model may satisfy benchmark-level forgetting criteria while still revealing target knowledge under stronger prompting, relearning, or recovery procedures. Therefore, SAUL should not be interpreted as providing a complete removal guarantee against all post-unlearning attacks. Its empirical results should be understood within the evaluated benchmarks, metrics, and threat models.

Another risk is over-forgetting. If the forgetting threshold is set too aggressively, an unlearning method may degrade benign capabilities or remove related non-target knowledge. This is especially important for LLMs, where target knowledge can be entangled with surrounding factual, linguistic, or domain knowledge. The constrained formulation in SAUL is designed to reduce unnecessary forgetting pressure, but appropriate threshold selection and post-unlearning evaluation remain necessary.

We do not release new sensitive datasets or model checkpoints as part of this work. Our experiments use existing research benchmarks and report aggregate evaluation results. Future deployments of unlearning methods should be accompanied by task-specific risk assessment, stronger adaptive recovery evaluations, and validation that the resulting model does not introduce unacceptable degradation on benign use cases.

H Use of AI Assistants

We used AI assistants for language editing, including grammar correction and improving the clarity of the manuscript. We also used GPT-based models to generate paraphrased variants of questions for the paraphrased-question evaluation. The generated paraphrases were manually reviewed to ensure that they remained semantically consistent with the original questions and were used solely as evaluation inputs, not for model training or model selection.