Learning What to Forget: Improving LLM Unlearning via Learned Token-Level Importance

Gizem Yüce* **Giorgos Nikolaou*** **Nicolas Flammarion**
Theory of Machine Learning Lab, EPFL
{gizem.yuce, georgios.nikolaou, nicolas.flammarion}@epfl.ch

Abstract

Machine unlearning aims to remove targeted knowledge from a trained model while preserving its general capabilities. For autoregressive language models, not all tokens in a forget sample are equally relevant to forgetting. Existing approaches either ignore this heterogeneity or rely on auxiliary models, heuristics, or external annotations to estimate each token's relevance for forgetting. We instead propose retain-conflict as a criterion for identifying forget-specific tokens, measuring how well a subset of tokens can be suppressed while preserving performance on the retain set. We formalize this principle as a fixed-budget joint optimization problem over model parameters and token weights, allowing token selection to be learned directly from the unlearning objective. Motivated by this formulation, we introduce **Alternating Token-Weighted Unlearning** (ATWU), a lightweight framework that jointly learns token forget-specificity and model parameters during unlearning using a simple linear scorer over hidden states, without external token-level supervision. Across TOFU, RWKU, and WMDP, ATWU achieves state-of-the-art forget–retain trade-offs, outperforming sample-level methods, probability-based token-weighting heuristics, and auxiliary-model-based approaches. Moreover, the learned scores align substantially better with ground-truth forget-specific spans, indicating that ATWU identifies semantically meaningful token-level forgetting signals. Overall, our results suggest that retain conflict provides an effective criterion for identifying what language models should forget, enabling unsupervised learning of token-level forget-specificity directly from model representations with minimal computational overhead.

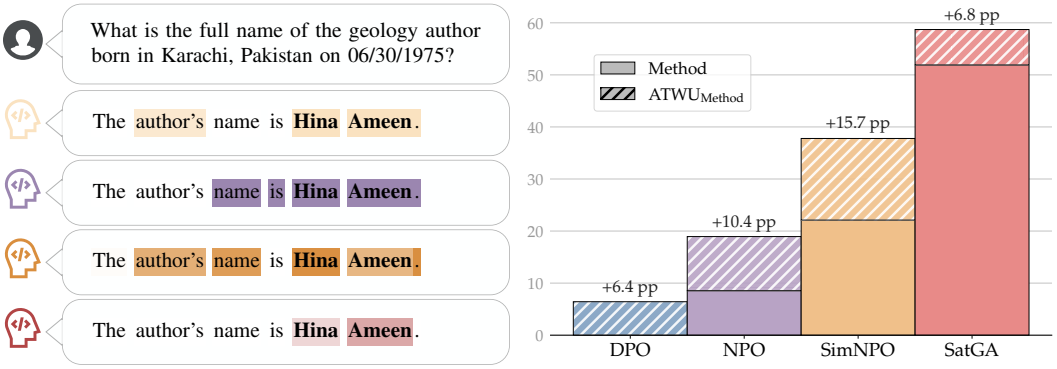
1 Introduction

Large language models trained on web-scale corpora can memorize personal information [Nasr et al., 2025], copyrighted text [Karamolegkou et al., 2023, Carlini et al., 2023], and harmful content [Li et al., 2024, Barrett et al., 2023]. However, the sheer scale of these models and their training data makes it infeasible to identify the specific training samples containing such information and retrain the model from scratch without them. Machine unlearning [Cao and Yang, 2015, Bourtoule et al., 2021, Nguyen et al., 2025] has therefore become a critical capability for deploying large language models responsibly, enabling the removal of sensitive, copyrighted, or harmful knowledge without retraining from scratch. Despite its importance, scalable and reliable unlearning remains far from solved, particularly in the context of modern language models.

Unlearning in large language models requires identifying which parts of a forget sample actually encode the targeted information to be removed. As in standard training, the unlearning objective decomposes over tokens in an autoregressive manner. However, when given a forget sample (e.g., a sentence or document to be unlearned), not all tokens contribute equally to the information that should

*Equal contribution

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) (b)

Figure 1: (a) Token-level scores on a TOFU forget sample. SEUL, SU-LLM, and SATIMP assign substantial weight to both forget-specific and structural tokens, whereas ATWU concentrates on the bold ground-truth forget-specific span. (b) ATWU can be combined with diverse forget losses. For each objective (DPO, NPO, SIMNPO, and SATGA) solid bars denote baseline Unlearning Quality (UQ), and added segments denote the gains from using ATWU token weights with the corresponding forget loss. ATWU consistently improves the forget–retain trade-off.

be removed. Many tokens such as common function words, punctuation, and syntactic patterns are *structural* and would be generated regardless of exposure to the forget data. Penalizing the generation of such tokens does not promote forgetting of the targeted content, but instead risks degrading general language capabilities.

This token heterogeneity has motivated token-weighted unlearning objectives [Yang et al., 2025, Wan et al., 2025, Wang et al., 2025a, Zhou et al., 2026], in which the forget loss is selectively applied to tokens judged to encode the targeted information. Existing approaches, however, still obtain these token weights from external annotations, auxiliary models, probability heuristics, or task-specific rules. In contrast, we characterize token-level forget-specificity through the interaction between the forget and the retain objectives themselves. While prior work typically treats retain conflict merely as a motivation for token-selective unlearning, we argue that it can also provide a criterion for selection. A forget token is deemed *forget-specific* if applying the forget objective to that token is compatible with remaining close to retain optimality, whereas applying the same objective to structural tokens induces retain degradation. This turns token weighting from an external heuristic into a latent variable of the unlearning objective.

Contributions. Our contributions are threefold:

- We propose retain conflict as a criterion for identifying forget-specific tokens and formulate token selection as a fixed-budget joint optimization problem over token weights and model parameters.
- We instantiate this principle with ATWU (**Alternating Token-Weighted Unlearning**), which learns token weights directly from the interaction between the forget and retain objectives using a lightweight scorer over the language model's hidden states, without requiring token-level annotations or auxiliary models.[2]
- Across TOFU, RWKU, and WMDP, we show that ATWU improves the forget–retain trade-off over sample-level baselines, probability-based token heuristics, and auxiliary-model-based token weighting, while producing token scores that align with ground-truth forget-specific spans.

2 Related Work

The problem of machine unlearning for LLMs has been approached through a variety of objectives. Gradient Ascent (GA) maximizes the cross-entropy loss on forget samples [Jang et al., 2023], while Gradient Difference (GRADDIFF) combines this ascent with retain-set descent or regularization [Liu

[2]Code is available at https://github.com/tml-epfl/ATWU.

et al., 2022]. Preference-based approaches include DPO-style refusal objectives [Rafailov et al., 2023], Negative Preference Optimization (NPO) [Zhang et al., 2024], and reference-free variants such as SIMNPO [Fan et al., 2025]. Other methods use different mechanisms: JENSUN employs Jensen–Shannon-divergence objectives for more stable unlearning dynamics [Singh et al., 2025], RMU perturbs representations associated with the forget set [Li et al., 2024], and UNDIAL distills toward adjusted output distributions [Dong et al., 2025].

Mavrothalassitis et al. [2025] show that statistical dependence between the forget and retain sets can fundamentally limit ascent–descent-based unlearning, causing updates intended to suppress forget knowledge to interfere with knowledge that should be retained. Token-weighted unlearning approaches aim to identify tokens that encode the targeted information and selectively apply the forgetting objective to avoid unnecessary utility degradation. This can also be viewed as a form of credit assignment, analogous to the problem in reinforcement learning for LLMs of attributing sequence-level rewards to individual token predictions [Zhang, 2026].

Token-weighted unlearning approaches typically adopt one of the sample-level objectives above for the actual forget update, and differ primarily in how token forget-specificity is estimated. *Auxiliary-model approaches* infer it by comparing the predictions of models fine-tuned on retain and forget splits [Eldan and Russinovich, 2023, Wan et al., 2025]. *Probability-based heuristics* instead use model confidence, entropy or other loss-based criteria as proxies for token forget-specificity [Wang et al., 2025b, Yang et al., 2025, Koh et al., 2026]. Other methods rely on external signals such as linguistic parsers [Dong et al., 2025] or LLM-generated annotations [Zhou et al., 2026] to identify forget-specific tokens. In particular, approaches based on large language model-generated annotations can provide strong supervision, but may be costly to scale and raise additional privacy concerns. In contrast, ATWU learns token-level weights jointly with the unlearning update, using the same retain and forget objectives that define the desired deletion behavior, and requires neither external supervision nor auxiliary model training.

Finally, our scoring mechanism is motivated by work showing that contextual language-model representations encode rich linguistic and semantic information and that many token-level properties are linearly recoverable from intermediate representations [Alain and Bengio, 2017, Belinkov et al., 2017, Tenney et al., 2019]. While this literature is primarily descriptive and does not address unlearning, it motivates parameterizing token forget-specificity as a lightweight linear scorer over hidden states.

3 Token-Weighted Unlearning as a Joint Optimization Problem

We now formalize the token-selection problem underlying ATWU. The goal is to define when a forget token should receive the forget update, and to show that this selector can be identified from the retain and forget objectives themselves.

3.1 Background and Notation

Let $\mathcal{V}$ be a finite vocabulary, $\boldsymbol{\Theta} \subseteq \mathbb{R}^d$ the parameter space, and $p_{\boldsymbol{\theta}} : \mathcal{V}^* \to \Delta(\mathcal{V})$ an autoregressive language model. We write $p_{\boldsymbol{\theta}}(\mathrm{x}_t \mid \mathrm{x}_{<t})$ for the probability of token x_t assigned by the model given the context $\mathrm{x}_{<t}$ and $|\mathrm{x}|$ for the length of sequence x. We are given a pretrained model $p_{\boldsymbol{\theta}_\mathrm{o}}$, a forget set $\mathcal{D}_\mathrm{f}$ whose influence should be removed, and a retain set $\mathcal{D}_\mathrm{r}$ on which model performance should be preserved.

The common framework for LLM unlearning combines a retain loss $\ell_\mathrm{r}(\mathrm{x}_t \mid \mathrm{x}_{<t}\,;\,\boldsymbol{\theta})$, which preserves behavior on retain samples, with a forget loss $\ell_\mathrm{f}(\mathrm{x}_t \mid \mathrm{x}_{<t}\,;\,\boldsymbol{\theta})$, which suppresses the original forget completion. The retain loss is typically cross-entropy ℓ_CE or KL divergence to the pretrained model predictions. Forget losses include negative cross-entropy $\ell_\mathrm{GA} = -\ell_\mathrm{CE} = \log p_{\boldsymbol{\theta}}(\mathrm{x}_t \mid \mathrm{x}_{<t})$ with various scalings and preference-based losses. For clarity, we absorb fixed loss coefficients and dataset normalizations into ℓ_r and ℓ_f. Finally, we write $R(\boldsymbol{\theta}) \coloneqq \sum_{\mathrm{x}\in\mathcal{D}_\mathrm{r}} \sum_{t=1}^{|\mathrm{x}|} \ell_\mathrm{r}(\mathrm{x}_t \mid \mathrm{x}_{<t}\,;\,\boldsymbol{\theta})$ for the retain objective and $R^\star \coloneqq \min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} R(\boldsymbol{\theta})$ for its minimum over the parameter space.

3.2 Forget-Specific vs. Structural Tokens

The forget set contains both information that should be removed and linguistic structure that should be preserved. Tokens such as names, dates, quantities, rare identifiers, and attribute-bearing phrases can identify the targeted knowledge in $\mathcal{D}_\mathrm{f}$; we call these tokens **forget-specific**. In contrast, function words, punctuation, syntactic patterns, and generic continuations are **structural**: they are useful under the retain distribution and would often be generated even without exposure to the forget data. Applying the forget loss to forget-specific tokens removes targeted information, while applying it to structural tokens can instead damage general language capabilities.

This distinction has motivated the introduction of token-level weights $z_{\mathrm{x},t} \in [0,1]$ approximating the oracle binary label $z^\star_{\mathrm{x},t} = \mathbf{1}_{\{\mathrm{x}_t \text{ is a forget-specific token}\}}$, and the token-weighted loss

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{z}) \;=\; \sum_{\mathrm{x}\in\mathcal{D}_\mathrm{r}} \sum_{t=1}^{|\mathrm{x}|} \ell_\mathrm{r}(\mathrm{x}_t \mid \mathrm{x}_{<t}\,;\,\boldsymbol{\theta}) \;+\; \sum_{\mathrm{x}\in\mathcal{D}_\mathrm{f}} \sum_{t=1}^{|\mathrm{x}|} z_{\mathrm{x},t}\, \ell_\mathrm{f}(\mathrm{x}_t \mid \mathrm{x}_{<t}\,;\,\boldsymbol{\theta}). \tag{1}$$

If the oracle labels $\mathbf{z}^\star$ were known, re-weighted unlearning could simply compute a minimizer $\boldsymbol{\theta}^\star \in \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathbf{z}^\star)$. This oracle setting is useful as an analysis tool—Table 4 later shows that using ground-truth token labels can indeed substantially improve the forget–retain trade-off—but such labels are unavailable or very costly in practice. Prior token-weighted methods therefore approximate $\mathbf{z}^\star$ using annotations, auxiliary models, or probability-based proxies such as confidence, saturation, and surprisal. These methods, which adopt the formulation in Equation 1, are likewise motivated by the retain conflict: minimizing the forget loss on structural tokens harms retain performance, while doing so on the forget-specific tokens does not. The key difference in our method from prior works is that we use retain conflict as a token-level identification criterion.

3.3 Retain Conflict as a Token-Selection Criterion

Let $\mathcal{I}_F := \{(x,t) : x \in \mathcal{D}_\mathrm{f},\, t \in [|x|]\}$ denote the set of forget sequence position pairs and $N_F := |\mathcal{I}_F|$ be the number of tokens in the forget set. For each $i = (x,t) \in \mathcal{I}_F$, write $\ell_\mathrm{f}^i(\boldsymbol{\theta}) = \ell_\mathrm{f}(\mathrm{x}_t \mid \mathrm{x}_{<t}\,;\,\boldsymbol{\theta})$. For any subset $\mathcal{A} \subseteq \mathcal{I}_F$, define the retain conflict

$$\kappa(\mathcal{A}) := \min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left[R(\boldsymbol{\theta}) - R^\star + \sum_{i\in\mathcal{A}} \left(\ell_\mathrm{f}^i(\boldsymbol{\theta}) - \ell_\mathrm{f}^{\min}\right) \right], \tag{2}$$

where $\ell_\mathrm{f}^{\min} := \min_{\boldsymbol{\theta},i} \ell_\mathrm{f}^i(\boldsymbol{\theta})$ is the global minimum of the forget loss.[3] Intuitively, $\kappa(\mathcal{A})$ measures the minimum residual cost of suppressing the tokens in $\mathcal{A}$ while preserving performance on the retain set. A set with low retain conflict can be jointly suppressed with little interference with the retain objective, whereas a set with high retain conflict cannot. Note that $\kappa(\mathcal{A})$ is non-negative and monotone, i.e., for $\mathcal{A} \subseteq \mathcal{B}$, $\kappa(\mathcal{A}) \le \kappa(\mathcal{B})$.

We propose using retain conflict as a criterion for selecting which token positions should receive the forget update. The effectiveness of this approach depends on whether the lowest-conflict subsets indeed correspond to semantically forget-specific tokens. Whether this holds in practice depends on the forget set and the model's capacity. Although it is not possible to completely validate this alignment in practice, we provide empirical evidence and discussion in Section E.1 that support retain conflict as a token-selection criterion.

Since κ is monotone, minimizing it without constraints would yield the empty set. We therefore consider selecting a fixed number ρN_F of tokens, where $\rho \in (0,1]$ controls the selection budget and ρN_F is assumed to be an integer. Moreover, directly identifying and using the minimum-conflict subset is computationally intractable, as evaluating $\kappa(\mathcal{A})$ requires optimizing the model parameters for each candidate subset.

Rather than evaluating the retain conflict of candidate subsets individually, we express this selection problem as a joint optimization over model parameters and binary token weights. Note that for any subset $\mathcal{A} \subseteq \mathcal{I}_F$ of size ρN_F, with indicator vector $\mathbf{z}_\mathcal{A}$, the definition of κ gives

$$\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{\theta}, \mathbf{z}_\mathcal{A}) = R^\star + \rho N_F \ell_\mathrm{f}^{\min} + \kappa(\mathcal{A}).$$

[3] We assume the forget loss is bounded below and that the minima defining $R^\star$, $\ell_\mathrm{f}^{\min}$, and κ are attained. If $\ell_\mathrm{f}^{\min}$ is only an infimum, all statements hold with $\min$ replaced by $\inf$.

Consequently, minimizing retain conflict at a fixed budget is equivalent to solving

$$\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta},\ \mathbf{z}\in\mathcal{Z}} \mathcal{L}(\boldsymbol{\theta},\mathbf{z}), \qquad \mathcal{Z} \coloneqq \left\{\mathbf{z}\in\{0,1\}^{N_F} : \textstyle\sum_{i\in\mathcal{I}_F} z_i = \rho N_F\right\}. \tag{3}$$

In practice, ρ serves as a scalar hyperparameter controlling the budget of this learned selector and we provide sensitivity analysis for our method in Section D.6.

4 Alternating Token-Weighted Unlearning

We now turn the joint formulation into a practical learning algorithm: *Alternating Token-Weighted Unlearning* (ATWU). We first relax the discrete optimization problem, then introduce the hidden-state scorer, specify and justify the selection of forget loss used in our experiments, and finally describe the alternating optimization procedure.

Penalized relaxation. The constrained problem (3) is combinatorial in $\mathbf{z}$. We therefore instead optimize a continuous penalized objective

$$\tilde{\mathcal{L}}(\boldsymbol{\theta},\mathbf{z}) = \mathcal{L}(\boldsymbol{\theta},\mathbf{z}) + \lambda_{\text{ent}} \sum_{z_{\mathrm{x},t}\in\mathbf{z}} H(z_{\mathrm{x},t}) + \lambda_{\text{pop}}\Big(\tfrac{1}{N_F}\sum_{z_{\mathrm{x},t}\in\mathbf{z}} z_{\mathrm{x},t} - \rho\Big)^2, \tag{4}$$

where $H(z) = -z\log z - (1-z)\log(1-z)$ is the binary entropy.

The objective $\tilde{\mathcal{L}}(\boldsymbol{\theta},\mathbf{z})$ relaxes the original combinatorial problem (3) on two fronts: the binary constraint $\mathbf{z}\in\{0,1\}^{N_F}$ is relaxed to $\mathbf{z}\in[0,1]^{N_F}$, and the budget constraint $\sum z_{\mathrm{x},t} = \rho N_F$ is replaced by the quadratic penalty. The entropy term counters the first relaxation by pushing each $z_{\mathrm{x},t}$ back towards $\{0,1\}$. This relaxation can be *exact* under sufficiently large regularization.

Lemma 4.1 (Exactness of the relaxation, informal)**.** *For sufficiently large λ_{ent} and λ_{pop}, every global minimizer of the relaxed objective (4) over $[0,1]^{N_F}$ is binary, satisfies the budget constraint, and is a global minimizer of the constrained problem (3).*

The formal statement and proof are given in Lemma E.1. While this free-token-weight formulation provides strong theoretical guarantees, learning an independent scalar for every token is unscalable for large corpora and cannot generalize to unseen sequences. To bridge this gap from theory to practice, ATWU parameterizes the selector using a shared scoring function $g_{\mathbf{w}}$. Under this scalable parameterization, the entropy and budget terms naturally transition from enforcing exact combinatorial constraints to acting as principled regularizers, successfully guiding the network toward sparse, budget-controlled token selection.

To instantiate this shared scorer, we parameterize token forget specificity using a simple linear projection over the language model's hidden representations. This architectural choice is directly motivated by the structural probing literature: because intermediate hidden states natively encode rich linguistic and semantic properties [Tenney et al., 2019], they provide a natural representation space for identifying forget-specific patterns without requiring a complex auxiliary network.

Let $\mathbf{h}_{\boldsymbol{\theta}}(\mathrm{x}_t)\in\mathbb{R}^d$ denote the t-th hidden representation produced by $p_{\boldsymbol{\theta}}$ for the sequence x. ATWU defines

$$\hat{z}_{\mathrm{x},t} = g_{\mathbf{w}}(\mathbf{h}_{\boldsymbol{\theta}}(\mathrm{x}_t)) = \sigma(\langle \mathbf{w}, \mathbf{h}_{\boldsymbol{\theta}}(\mathrm{x}_t)\rangle) \in (0,1), \tag{5}$$

where $\mathbf{w}\in\mathbb{R}^d$ is the learned scorer parameter.

This single linear projection keeps the mechanism lightweight while encouraging reusable token-level patterns rather than independent per-token decisions. We initialize $\mathbf{w} = \mathbf{0}_d$, meaning all tokens initially receive a uniform score of $1/2$. Consequently, the objective begins token-agnostic and smoothly becomes selective during training. As a proof of concept, Appendix D.1 shows that this exact linear architecture can recover ground-truth forget labels when trained with explicit supervision, i.e., the scorer model has enough capacity to model the true forget specificity; ATWU, however, learns the scores entirely implicitly from the unlearning objective.

Choice of loss functions. We use standard cross-entropy as the retain loss and a saturated negative cross-entropy introduced by Wang et al. [2025b] as the base forget loss. For a hyperparameter $\beta > 0$, the token-wise forget loss is defined as:

$$\ell_{\mathrm{f}}(\mathrm{x}_t \mid \mathrm{x}_{<t}\,;\,\boldsymbol{\theta}) \ \coloneqq\ p_{\boldsymbol{\theta}}(\mathrm{x}_t \mid \mathrm{x}_{<t})^{\beta}\cdot \log p_{\boldsymbol{\theta}}(\mathrm{x}_t \mid \mathrm{x}_{<t}). \tag{6}$$

This loss formulation incorporates the saturation weighting mechanism utilized by Wang et al. [2025b] inside the loss, which we refer to as SATGA. While saturation alone is a poor proxy for token-level forget-specificity (as evidenced by the poor AUROC of the saturation score in Fig. 2), it provides highly desirable gradient scaling properties for the loss function.

Crucially, unlike raw negative cross-entropy ℓ_{GA}, this saturated variant is bounded below. As a function of $p \in (0, 1]$, the term $p^\beta \log p$ reaches a minimum of $-1/(e\beta)$ at $p = e^{-1/\beta}$. Consequently, the hyperparameter β effectively controls the target probability level for the forget update, with smaller β values enforcing stronger suppression.

To instantiate our general joint-learning framework using this bounded objective, we introduce a slight modification, where we inject the dynamically learned token score into the saturation exponent. We refer to this score-modulated variant as SATGA$^+$. Plugging this into the joint formulation yields our primary ATWU objective:

$$\begin{aligned}\mathcal{L}_{\text{ATWU}}(\boldsymbol{\theta}, \mathbf{w}) = \alpha R(\boldsymbol{\theta}) + \gamma \sum_{\mathrm{x},t} g_{\mathbf{w}}(\mathrm{x}_t) \cdot p_{\boldsymbol{\theta}}(\mathrm{x}_t \mid \mathrm{x}_{<t})^{\beta g_{\mathbf{w}}(\mathrm{x}_t)} \cdot \log p_{\boldsymbol{\theta}}(\mathrm{x}_t \mid \mathrm{x}_{<t}) \\ + \lambda_{\text{ent}} \sum_{\mathrm{x},t} H\big(g_{\mathbf{w}}(\mathrm{x}_t)\big) + \lambda_{\text{pop}} \Big(\tfrac{1}{N_F} \textstyle\sum_{\mathrm{x},t} g_{\mathbf{w}}(\mathrm{x}_t) - \rho \Big)^2,\end{aligned} \tag{7}$$

where $g_{\mathbf{w}}(\mathrm{x}_t) \coloneqq g_{\mathbf{w}}(\mathbf{h}_{\boldsymbol{\theta}}(\mathrm{x}_t))$.

For strictly binary scores, the SATGA$^+$ modification reduces to standard SATGA. However, because our scores $g_{\mathbf{w}}$ are continuous during training, the modified exponent forces uncertain tokens to exert a much smoother, attenuated forget update. This stabilizes the early phases of optimization before the scorer has fully converged, an effect we ablate in Table 4. The coefficients α and γ denote the retain and forget loss weights, respectively. While we absorbed these constants into the base loss definitions in earlier sections for notational simplicity, we make them explicit here to reflect our exact empirical objective.

Alternating optimization. We optimize $\mathcal{L}_{\text{ATWU}}(\boldsymbol{\theta}, \mathbf{w})$ by alternating between language-model and scorer updates. With the scorer $\mathbf{w}$ fixed, the language-model parameters $\boldsymbol{\theta}$ are updated using the current token-weighted forget objective together with the retain objective. With $\boldsymbol{\theta}$ fixed, the scorer $\mathbf{w}$ is updated to improve token selection under the same regularized objective. During model updates, the scores are detached and treated as fixed coefficients; during scorer updates, $\boldsymbol{\theta}$ is frozen and gradients flow only through $\mathbf{w}$. This scheduled alternation is empirically more stable than updating the scorer and model in lockstep. The scorer changes the effective forget objective, while the model changes the hidden-state geometry on which the scorer depends. Updating them on separate timescales reduces this feedback loop. Finally, in Appendix D.5 we present an ablation on the update frequency that shows that the alternating variant outperforms joint updates.

5 Experiments

In this section, we demonstrate that the token-level weights learned by ATWU translate into better unlearning while preserving model utility, and analyze what drives these gains.

Benchmarks and models. We evaluate ATWU on three LLM unlearning benchmarks covering different types of targeted knowledge. TOFU [Maini et al., 2024] is a synthetic question-answering benchmark built from fictitious-author biographies. RWKU [Jin et al., 2024] targets real-world knowledge about public figures in a pretrained LLM. Following Singh et al. [2025], we perform batch unlearning on a fixed set of ten subjects. WMDP [Li et al., 2024] evaluates the removal of harmful knowledge in biology and cybersecurity, providing a substantially different forgetting setting from the biographical and entity-level knowledge in TOFU and RWKU. On TOFU, we use `Llama-3.1-8B-Instruct` and `Llama-3.2-1B-Instruct` [Grattafiori et al., 2024], using the checkpoints released by Dorna et al. [2025]. On RWKU, we use `Phi-3-Mini-4k-Instruct` [Abdin et al., 2024], and on WMDP we use `Zephyr-7B-beta` [Tunstall et al., 2023]. Hardware, software, and optimization details used across all settings are provided in Appendix C.

Hyperparameter tuning. For each method and benchmark, we tune method-specific hyperparameters using Bayesian optimization [Akiba et al., 2019] under a common tuning protocol with

Method		Unlearning			Utility				Unlearning			Utility		
		FQ ↑	RD ↓	UQ ↑	MMLU	Rep.	WR		FQ ↑	RD ↓	UQ ↑	MMLU	Rep.	WR
ORIGINAL		0.0	0.0	0.0	66.6	546	50.0		0.0	0.0	0.0	70.1	529	50.0
GRADDIFF	TOFU	39.7	17.3	22.4	66.1	545	48.0	RWKU	<u>81.9</u>	36.7	45.2	69.2	514	43.0
DPO	TOFU	59.5	15.0	44.5	64.3	533	38.5	RWKU	12.3	**17.0**	0.0	69.7	523	43.0
NPO	TOFU	48.0	5.0	43.0	66.2	551	48.5	RWKU	47.6	39.1	8.5	68.1	543	33.5
SIMNPO	TOFU	69.3	6.5	62.8	65.8	541	50.5	RWKU	51.8	29.7	22.1	68.8	529	45.5
JENSUN	TOFU	**98.0**	9.6	<u>88.3</u>	65.3	236	11.0	RWKU	**85.4**	36.1	49.3	70.2	523	46.5
RMU	TOFU	87.2	**1.3**	85.9	65.3	540	50.5	RWKU	—	—	—	—	—	—
WGA	TOFU	66.8	4.8	62.1	66.0	550	60.0	RWKU	78.0	26.1	<u>51.9</u>	69.4	518	41.0
SATIMP	TOFU	78.6	<u>3.0</u>	75.6	65.9	549	53.5	RWKU	77.3	29.3	48.0	69.8	518	40.0
ATWU	TOFU	<u>95.2</u>	3.5	**91.7**	66.5	558	58.0	RWKU	81.4	<u>22.7</u>	**58.7**	70.3	519	46.0

FQ: forget quality, the relative reduction in worst-case forget-set judge score. RD: retain degradation, the relative loss in retain-set judge score. $\mathrm{UQ} = [\mathrm{FQ} - \mathrm{RD}]_+$: net forget–retain trade-off. Definitions in Appendix B.6.

Table 1: ATWU achieves the best UQ on both benchmarks while preserving utility close to the original checkpoint, the only method to do so consistently across the six metric panels. Left: TOFU `forget10` with `Llama-3.1-8B-Instruct`; right: canonical RWKU ten-subject batch with `Phi-3-Mini-4k-Instruct`. Higher-FQ competitors such as JENSUN forget aggressively but collapse generation quality; RMU matches ATWU on retain degradation but trails on UQ. **Best** (**bold**) and <u>second-best</u> (<u>underlined</u>) per column; utility values that are materially degraded (MMLU drop $>$ 2pp, Rep. drop $> 6\%$, or WR < 45) are shown in red.

benchmark-specific surrogate metrics that are inexpensive to evaluate. Search spaces, tuning budgets, and selected configurations are provided in Appendices C.3 and C.4.

Evaluation. Singh et al. [2025] show that apparent forgetting under the original query or under simple evaluation metrics may not persist under perturbed inputs. Following their robustness-oriented semantic evaluation, we use `GPT-5.4 mini` as an LLM judge. Given a question, its reference answer, and the model response, the judge makes a binary semantic decision indicating whether the response recovers the reference answer. On the forget set, we evaluate multiple paraphrases of each question and additionally test whether the forgotten answer resurfaces when the model is primed with related in-context demonstrations. For each sample on the forget set, we take the worst-case score across these evaluations, so forgetting is deemed successful only when the target information remains unrecoverable under both paraphrasing and in-context relearning. On the retain set, we instead average the judge score across paraphrases to measure how reliably retained information is preserved. The resulting judge scores are aggregated into forget quality (FQ), which measures the relative reduction in worst-case recoverability of the targeted information after unlearning; retain degradation (RD), which measures the relative reduction in the model's ability to recover retain information; and unlearning quality $\mathrm{UQ} = [\mathrm{FQ} - \mathrm{RD}]_+$, which summarizes the improvement in forgetting after accounting for degradation on retain performance. We validate the judge against human annotations on 440 responses, obtaining 96.6% agreement.

For RWKU, this evaluation includes all three native forget-probe categories: open question answering (QA), fill-in-the-blank (FB), and adversarial attacks (AA), with the latter including prefix injection, role-playing, reverse querying, affirmative suffixes, and cross-lingual variants. The FQ and UQ results aggregate these probe categories; we report the QA/FB/AA breakdown separately in Table 13.

Unlike metrics based on token-overlap with the original answers or likelihood-based surrogates, the judge-based protocol penalizes methods that suppress only the original surface form while leaving semantically equivalent completions recoverable. We complement these judge-based metrics with three utility-preservation metrics: MMLU accuracy, generation repetitiveness (Rep.), and pairwise win rate against a reference model (WR; the retain model on TOFU, the original checkpoint otherwise). For WMDP, we follow the benchmark's multiple-choice evaluation and report accuracy on the biology and cybersecurity subsets, where lower accuracy indicates stronger forgetting. We report the same three utility measures and, since WMDP evaluation does not require the semantic LLM judge, repeat each selected configuration over three unlearning seeds and report mean and standard deviation. Full metric definitions are provided in Appendix B, and further experimental details in Appendix C.

Method	Token weights $z_{\mathrm{x},t}$	Surrogate			Relative			Utility		
		$\mathrm{ES}_{\mathcal{F}}\downarrow$	$\mathrm{ES}_{\mathcal{R}}\uparrow$	$\mathrm{ES}_{\Delta}\uparrow$	FQ $\uparrow$	RD $\downarrow$	UQ $\uparrow$	MMLU	Rep.	WR
ORIGINAL	—	0.706	0.737	0.030	0.0	0.0	0.0	45.1	559	50.0
SEUL	logprob mask	0.101	0.675	0.574	42.6	6.9	35.6	45.4	559	59.5
SU-LLM	LLM diff	0.190	0.704	0.514	31.3	**4.1**	27.2	45.2	564	56.0
SU-NGRAM	n-gram diff	0.074	0.634	0.559	55.4	15.8	39.7	45.3	560	51.5
FUNDIAL	noun mask	0.047	**0.759**	<u>0.712</u>	40.9	<u>5.6</u>	35.4	45.2	562	53.5
ETW	entropy	0.108	0.679	0.572	54.1	9.3	44.8	45.5	561	54.5
WGA	saturation	0.062	**0.759**	0.696	50.0	8.5	41.5	45.0	568	56.0
SATIMP	importance	<u>0.043</u>	0.747	0.704	<u>71.2</u>	**4.1**	<u>67.1</u>	44.8	570	55.0
ATWU	$g_{\mathbf{w}}(\mathrm{x}_t)$	**0.035**	<u>0.753</u>	**0.717**	**84.4**	6.3	**78.1**	45.0	575	51.5

SATIMP and ATWU use the SATGA and SATGA$^+$ forget losses, FUNDIAL a distillation loss, and all others negative cross-entropy (which WGA's saturation weights turn into SATGA).

Table 2: ATWU outperforms other token-weighted unlearning approaches on TOFU `forget10` (`Llama-3.2-1B-Instruct`). **Best** and <u>second-best</u> methods are highlighted.

5.1 Unlearning Quality

Table 1 and **??** compares ATWU with sample-level and token-weighted unlearning methods on TOFU and RWKU. Across both benchmarks, ATWU achieves the strongest overall unlearning quality while preserving general model utility. Several competing methods can remove the targeted information aggressively, but do so at the cost of retained knowledge or generation quality. ATWU instead improves forgetting without relying on broad degradation of the model, supporting our premise that restricting the forget update to the relevant token positions can reduce collateral damage. The advantage holds on `forget05` and across model scales, but shrinks on the smallest split, `forget01` (Tables 9 and 10; see Section 6).

We additionally evaluate on WMDP to test whether this behavior extends beyond biographical and entity-level knowledge. On `Zephyr-7B-beta`, ATWU reduces performance on the targeted biology and cybersecurity knowledge to near chance while largely preserving the utility measures considered here. Its forgetting performance is comparable to the strongest competing method, while exhibiting a more favorable utility profile. These results suggest that the benefits of selective token weighting extend to a substantially different type of forget target and model family. Full results are reported in Table 14, and the WMDP setup is described in Appendix C.2.

Compatibility with other forget objectives. So far, we have instantiated ATWU using a saturated negative cross-entropy loss. However, our framework of jointly learning the model and the scorer through alternating updates is loss-agnostic. Any forget loss that decomposes autoregressively over the tokens (Eq. 1) can be seamlessly plugged into the formulation. Moreover, this flexibility extends to forget objectives constructed from an autoregressive building block—such as the token-wise NLL of the forget sequence, or its log-ratio against a frozen reference—even if they do not perfectly match the exact formulation in Eq. 1. In such cases, we simply replace the autoregressive block with its scorer-reweighted analog, leaving the rest of the loss untouched. We apply this construction to DPO, NPO, and SIMNPO and provide further implementation details in Appendix D.4. Figure 1b and Table 3 demonstrate the unlearning quality of ATWU instantiated with these losses and the relative gain compared to the non-token weighted baselines. The absolute performance still depends strongly on the underlying forget loss, but the improvement from applying ATWU is consistent across objectives.

Method	Relative		Utility		
	UQ $\uparrow$	ΔUQ $\uparrow$	MMLU	Rep.	WR
ORIGINAL	0.0	—	70.1	529	50.0
$\mathrm{ATWU}_{\mathrm{DPO}}$	6.4	+6.4	69.5	512	36.0
$\mathrm{ATWU}_{\mathrm{NPO}}$	18.9	+10.4	67.7	545	35.5
$\mathrm{ATWU}_{\mathrm{SimNPO}}$	<u>37.8</u>	+15.7	69.9	531	43.5
ATWU	**58.7**	+6.8	70.3	519	46.0

Table 3: RWKU ten-subject batch results of ATWU with various forget losses. ΔUQ reports the gain over each method's vanilla counterpart in Table 1.

5.2 What makes ATWU effective?

ATWU learns semantically meaningful token scores. We first evaluate the learned scores independently of their effect on unlearning. Using the ground-truth token labels provided by Zhou et al. [2026], we compute AUROC for each sample in TOFU. Figure 2 shows the resulting distribution of per-sample AUROC scores across token-weighting and selection methods. ATWU aligns most strongly with the ground-truth annotations despite receiving no token-level supervision during training, while heuristics based on model confidence, entropy, saturation, or auxiliary-model disagreement provide substantially weaker signals. This supports our central hypothesis that the interaction between the forget and retain objectives provides a useful signal for identifying forget-specific token positions. Figure 11 additionally reports the pooled token-level ROC curves for these methods. Among the token-weighted approaches, ATWU also achieves the strongest overall unlearning performance, as shown in Table 2.

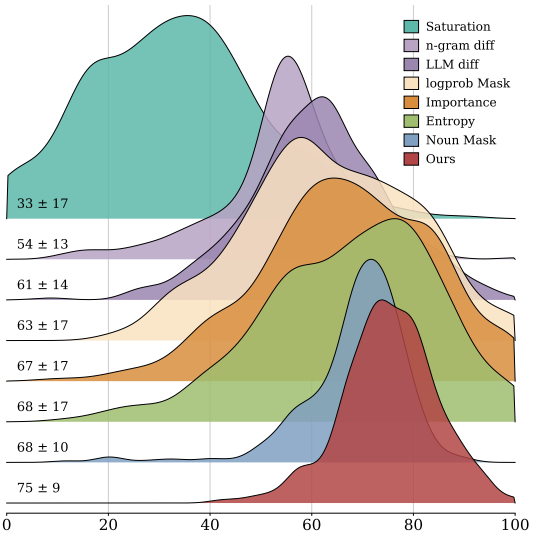


Figure 2: Per-sample AUROC distributions for token-level forget-specificity on TOFU forget10. Labels report mean ± std across forget samples.

Token selection and the forget objective play complementary roles. The comparison in Table 2 also reveals that better token identification does not automatically imply better unlearning. The methods differ not only in how they estimate token-level forget-specificity, but also in the forget objective to which these weights are applied. FUNDIAL uses its distillation-based objective, SATIMP and ATWU use SATGA and SATGA^{+}, respectively, while the remaining token-weighting approaches use token-wise negative cross-entropy. In particular, WGA's saturation weights turn this into the SATGA objective without informative token selection, and its saturation-based ranking is in fact worse than random at identifying forget-specific tokens. Nevertheless, it outperforms several methods with substantially better token rankings that use naive gradient ascent loss. Thus, score quality and unlearning quality are not interchangeable. Token weighting determines where the forget update is applied, while the forget objective determines how aggressively those selected positions are modified. In particular, the saturation term limits increasingly large updates as the target-token probability decreases, helping to avoid excessive forgetting even when token selection is imperfect.

A separate observation in Table 2 concerns evaluation. FUNDIAL performs similarly to ATWU under the extraction-based surrogate ES_{Δ}, yet the two methods differ substantially under the robustness-oriented judge-based UQ. This discrepancy illustrates why surrogate or surface-level extraction metrics alone can overstate forgetting, and motivates the semantic robustness evaluation used for our main comparisons.

Disentangling the score quality and forget loss. We further isolate these two factors by combining either ground-truth token labels or the learned ATWU scores with different forget objectives in Table 4. The oracle labels provide perfect token selection, yet pairing them with naive gradient ascent still produces a relatively poor forget–retain trade-off. Replacing gradient ascent with the saturated objective yields a much stronger result, showing that accurate token identification alone cannot compensate for an unsuitable forget loss. Conversely, with the same saturated objective, the learned ATWU scores substantially improve over unweighted updates, and GT scores improve over ATWU, confirm-

Score	Loss	Relative			Utility		
		FQ ↑	RD ↓	UQ ↑	MMLU	Rep.	WR
ORIGINAL	—	0.0	0.0	0.0	45.1	559	50.0
GT	GA	47.4	7.9	39.5	45.0	563	54.5
	SATGA	**90.2**	3.9	**86.2**	44.9	566	51.0
ATWU	GA	31.0	3.8	27.2	45.3	565	55.5
	SATGA	61.9	**3.4**	58.5	45.1	569	52.0
	SATGA^{+}	84.4	6.3	78.1	45.0	575	51.5

Table 4: Oracle score and weighting-scheme ablation on TOFU `forget10` (`Llama-3.2-1B-Instruct`).

ing that saturation alone is also insufficient. Finally, SATGA$^+$ exploits the continuous nature of the learned scores by placing them in the saturation exponent, attenuating the update on uncertain tokens while the scorer is still evolving. In conclusion, these results indicate that effective token-weighted unlearning requires both informative token scores and a forget objective that uses them appropriately.

6 Limitations

Despite its promising results, ATWU has several limitations. The retain-conflict criterion becomes less informative when the set size is very small. In this regime, an overparameterized model may be able to suppress many different token subsets while preserving retain performance by memorizing them, including subsets containing largely structural content. The desired forget-specific subset may therefore be only weakly distinguished from alternative low-conflict solutions. This is consistent with the weaker gains we observe on the smallest TOFU splits (see Appendix C.5). In addition, ATWU requires specifying the token-selection budget ρ, which is unknown in practice and is treated as a hyperparameter. Furthermore, our theoretical characterization concerns the ideal fixed-budget token-selection problem, whereas practical ATWU parameterizes token scores with a shared hidden-state scorer and learns them through alternating optimization. Consequently, even when the minimum-retain-conflict subset is unique, the practical method is not guaranteed to recover it due to both the restricted scorer parameterization and the non-convexity of the optimization problem.

Empirically, our experiments cover three benchmarks and three model families, and the judge-based TOFU and RWKU evaluations are single-run due to compute and judge-model cost, while WMDP results average three seeds. Finally, our primary semantic evaluation relies on an LLM judge. Although validated against human annotations, it remains an imperfect and potentially model-dependent measurement tool.

7 Conclusion

We introduced ATWU, reframing token-weighted unlearning as an end-to-end joint optimization problem rather than a pipeline with externally defined heuristics. By formalizing token specificity through retain conflict, we demonstrated that the unlearning objective itself contains sufficient latent signal to recover forget-specific tokens without any external supervision.

Across TOFU, RWKU, and WMDP, ATWU improves the forget–retain trade-off while largely preserving general model utility. Our ablations further show that accurate token identification alone is not sufficient for effective unlearning. The forget objective plays an equally important role, and the strongest results arise when informative token scores are paired with an objective that uses them appropriately.

Finally, explicitly annotated token labels can provide a useful supervised upper bound, but obtaining such annotations is costly and difficult to scale, and may be impractical in settings where the forget data itself is sensitive. ATWU avoids this dependence on external supervision. By recovering much of the benefit of oracle token selection using only representations from the model being unlearned, our results show that targeted token-level unlearning can be achieved without requiring annotated token labels or auxiliary models.

Acknowledgments

The authors would like to thank Francesco Croce for helpful discussions. This work was partially funded by an unrestricted gift from Coefficient Giving, and the grant number 212111 from the Swiss National Science Foundation. Gizem Yüce is supported by the Swiss AI Fellowship.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, et al. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219. 6, 24

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019. 6, 26

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2017. URL https://openreview.net/forum?id=ryF7rTqgl. 3

Clark Barrett, Brad Boyd, Elie Bursztein, Nicholas Carlini, Brad Chen, Jihye Choi, Amrita Roy Chowdhury, Mihai Christodorescu, Anupam Datta, Soheil Feizi, Kathleen Fisher, Tatsunori Hashimoto, Dan Hendrycks, Somesh Jha, Daniel Kang, Florian Kerschbaum, Eric Mitchell, John Mitchell, Zulfikar Ramzan, Khawaja Shams, Dawn Song, Ankur Taly, and Diyi Yang. Identifying and mitigating the security risks of generative AI. *Foundations and Trends® in Privacy and Security*, 6(1):1–52, December 2023. ISSN 2474-1566. doi: 10.1561/3300000041. URL http://dx.doi.org/10.1561/3300000041. 1

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. What do neural machine translation models learn about morphology? In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1080. URL https://aclanthology.org/P17-1080/. 3

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159, 2021. doi: 10.1109/SP40001.2021.00019. 1

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy*, SP '15, page 463–480, USA, 2015. IEEE Computer Society. ISBN 9781467369497. doi: 10.1109/SP.2015.35. URL https://doi.org/10.1109/SP.2015.35. 1

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TatRHT_1cK. 1

Yijiang River Dong, Hongzhou Lin, Mikhail Belkin, Ramon Huerta, and Ivan Vulić. UNDIAL: Self-distillation with adjusted logits for robust unlearning in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8827–8840, 2025. 3

Vineeth Dorna, Anmol Reddy Mekala, Wenlong Zhao, Andrew McCallum, J Zico Kolter, Zachary Chase Lipton, and Pratyush Maini. OpenUnlearning: Accelerating LLM Unlearning via Unified Benchmarking of Methods and Metrics. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Gy67Zh5X1i. 6, 24, 28, 32

Ronen Eldan and Mark Russinovich. Who's Harry Potter? Approximate unlearning in LLMs. arXiv, October 2023. URL https://www.microsoft.com/en-us/research/publication/whos-harry-potter-approximate-unlearning-in-llms/. 3

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity Prevails: Rethinking Negative Preference Optimization for LLM Unlearning. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=JbvSQm5h1l. 3, 40

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602. 21, 22, 30

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 6, 24

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *The Ninth International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ. 21

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 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, 2023. 2

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. RWKU: Benchmarking real-world knowledge unlearning for large language models. In *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=wOmtZ5FgMH. 6, 19, 21, 24, 25

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7403–7412, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.458. URL https://aclanthology.org/2023.emnlp-main.458/. 1

Seunghee Koh, Sunghyun Baek, Youngdong Kim, and Junmo Kim. Forget what matters, keep the rest: Selective unlearning of informative tokens, 2026. URL https://arxiv.org/abs/2604.17785. 3

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, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam Alfred Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar, Uday Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=xlr6AUDuJz. 1, 3, 6, 22, 24, 31

Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In Sarath Chandar, Razvan Pascanu, and Doina Precup, editors, *Proceedings of The 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, pages 243–254. PMLR, 2022. URL https://proceedings.mlr.press/v199/liu22a.html. 2

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=B41hNBoWLo. 6, 19, 24

Ioannis Mavrothalassitis, Pol Puigdemont, Noam Itzhak Levi, and Volkan Cevher. Ascent fails to forget. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=KBJSV1XApq. 3

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vjel3nWP2a. 1

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Trans. Intell. Syst. Technol.*, 16(5), September 2025. ISSN 2157-6904. doi: 10.1145/3749987. URL https://doi.org/10.1145/3749987. 1

OpenAI. GPT-5.4 mini, 2026. URL https://developers.openai.com/api/docs/models/gpt-5.4-mini. OpenAI API documentation. 30

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 35

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9. 3, 39

Naman Deep Singh, Maximilian Müller, Francesco Croce, and Matthias Hein. Unlearning that lasts: Utility-preserving, robust, and almost irreversible forgetting in LLMs, 2025. URL https://arxiv.org/abs/2509.02820. 3, 6, 7, 20, 21, 22, 24, 26, 32

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452/. 3, 5

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of LM alignment, 2023. URL https://arxiv.org/abs/2310.16944. 6, 24, 25

Yixin Wan, Anil Ramakrishna, Kai-Wei Chang, Volkan Cevher, and Rahul Gupta. Not every token needs forgetting: Selective unlearning balancing forgetting and utility in large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1827–1835, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.96. URL https://aclanthology.org/2025.findings-emnlp.96/. 2, 3

Lingzhi Wang, Xingshan Zeng, Jinsong Guo, Kam-Fai Wong, and Georg Gottlob. Selective forgetting: Advancing machine unlearning techniques and evaluation in language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 843–851, 2025a. 2

Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q Weinberger. Rethinking LLM unlearning objectives: A gradient perspective and go beyond. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=huo8MqVH6t. 3, 5, 6, 19, 31

Puning Yang, Qizhou Wang, Zhuo Huang, Tongliang Liu, Chengqi Zhang, and Bo Han. Exploring criteria of loss reweighting to enhance LLM unlearning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=mGOugCZlAq. 2, 3, 27, 31

Chenchen Zhang. From reasoning to agentic: Credit assignment in reinforcement learning for large language models, 2026. URL https://arxiv.org/abs/2604.09459. 3

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=MXLBXjQkmb. 3, 39

Xiangyu Zhou, Yao Qiang, Saleh Zare Zade, Douglas Zytko, Prashant Khanduri, and Dongxiao Zhu. Not all tokens are meant to be forgotten. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 38173–38182, 2026. 2, 3, 9, 36, 37, 38, 42, 43, 44

Appendix

Appendix Overview

A Qualitative Examples **16**

B Unlearning Metrics **19**

B.1 Token-level Extraction Metrics 19

B.2 Robustness-Aware Forget and Retain Quality 20

B.3 Utility Metrics 21

B.4 Native RWKU Benchmark Metrics 21

B.5 WMDP Metrics 22

B.6 Baseline-Relative Summary Metrics 22

C Experimental Details **24**

C.1 Benchmarks and Base Models 24

C.2 WMDP Unlearning Corpora 24

C.3 Hyperparameter Tuning Protocol 26

C.4 Hyperparameter Search Ranges and Results 27

C.5 Final Unlearning Runs and Evaluation 28

C.6 Computational Cost 34

C.7 Sensitivity to the Judge Model and Rubric 35

D Additional Experiments **36**

D.1 Linear Separability of Informative Tokens 36

D.2 Scorer Generalization to Held-Out Forget Data 37

D.3 Forget-Set Size and Scorer Learning 38

D.4 ATWU with Different Forget Losses 38

D.5 Ablations 40

D.6 Sensitivity to the Budget ρ and Saturation β 42

E Additional Analysis of the Joint Optimization Formulation **44**

E.1 Measuring Retain Conflict on TOFU 44

E.2 Proof of Lemma 4.1 (Formal Statement) 44

A Qualitative Examples

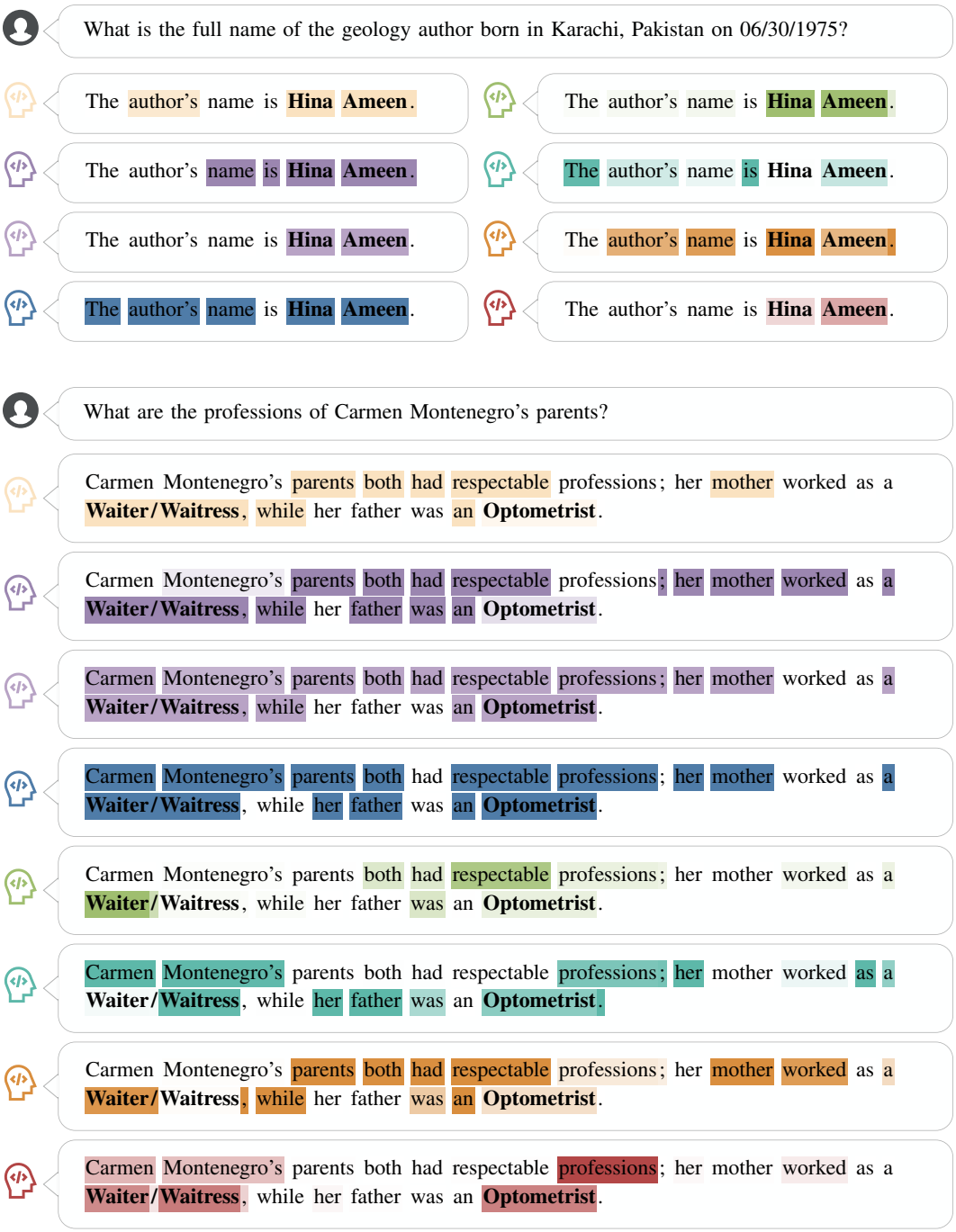


Figure 3: Token-level forget-specificity from SEUL, SU-LLM, SU-NGRAM, FUNDIAL, ETW, WGA, SATIMP, and ATWU on two TOFU forget samples. Shading reflects each method's raw token score; bold spans mark the ground-truth forget-specific tokens; ATWU concentrates most clearly on the answer-bearing spans.

We complement the quantitative results with qualitative examples of the token-level scores learned by ATWU. The goal is to inspect whether the scorer identifies the tokens that actually carry the forget target, rather than assigning high weight to generic context, function words, or syntactic scaffolding.

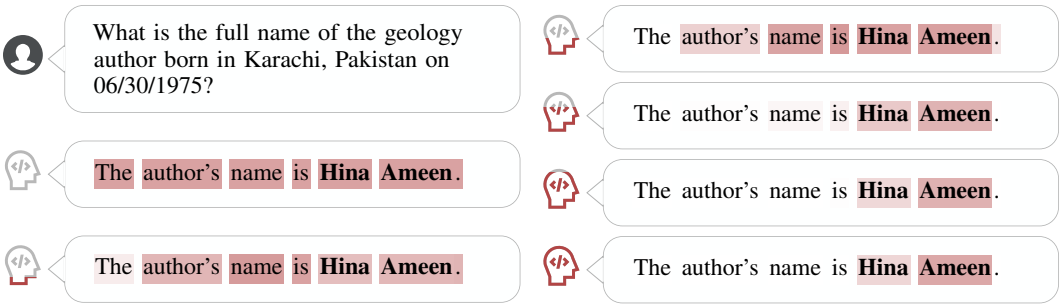


Figure 4: Evolution of ATWU's token scores during training. The scorer starts from uniform scores and gradually concentrates on the ground-truth forget-specific span.

In each example below, bold tokens indicate the ground-truth forget-specific span, and token shading indicates the importance score assigned by the corresponding method.

Figure 3 compares ATWU with several token-importance baselines on two TOFU forget samples. The first example is a short answer in which the forget-specific content is the author name, while the second is a longer answer in which the forget-specific content consists of the parents' professions. The binary scorers—SEUL, SU-LLM, SU-NGRAM, and FUNDIAL—tend to mark large portions of the answer as important, including tokens that are not specific to the forgotten fact. Probability-based heuristics such as WGA and SATIMP are less binary but often place substantial mass on structural tokens and surrounding context. ETW, which weights by token-level prediction entropy, identifies the answer name cleanly on the short example, but on the longer one it spreads mass over model-uncertain tokens that are not forget-specific (e.g., *respectable*, *both*, *had*). In contrast, ATWU assigns most of its mass to the answer-bearing spans while keeping generic tokens relatively low. This behavior is especially visible in the longer example, where the scorer separates the profession tokens from the surrounding sentence template.

Figure 4 visualizes how the ATWU scorer evolves during training on the short TOFU example. At initialization, all tokens receive the same score, so the forget loss is effectively token-uniform. As training progresses, the scorer first assigns nontrivial mass to several tokens in the answer, but gradually suppresses structural tokens and concentrates on the ground-truth forget-specific span. This trajectory illustrates the intended behavior of the alternating optimization: token selectivity is not imposed by external labels, but emerges as the scorer and unlearned model are optimized together.

Figure 5 summarizes this trend quantitatively by tracking token-ranking AUROC over training on TOFU `forget10`. The progression shows that the learned scores become increasingly aligned with the ground-truth forget-specific tokens over the course of optimization. This provides additional evidence that the scorer is not merely fitting token frequency or likelihood artifacts, but is learning a token-level signal that better matches the information to be removed.

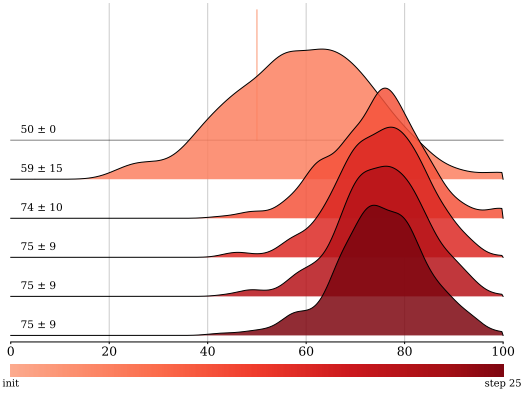


Figure 5: Progression of the per-sample AUROC distribution of ATWU's learned token scores during training on TOFU.

Finally, Figure 6 tests whether the learned scorer transfers beyond the exact forget samples used during training. We construct two unseen examples by taking TOFU-style questions and replacing the fictional entities with real people and their corresponding factual attributes. Although these examples are not part of the scorer's training data, ATWU still assigns elevated scores to the answer-bearing tokens—the name and sport tokens (*Michael*, *Phelps*, *swimming*) in the Olympic example, and the pet-identifying tokens (*Water Dogs*, *Bo*, *Sunny*) in the Obama example—while leaving most filler and syntactic tokens low. This indicates that the scorer is not merely memorizing the training entities or surface templates, but learns a transferable token-level signal for identifying forget-specific content.

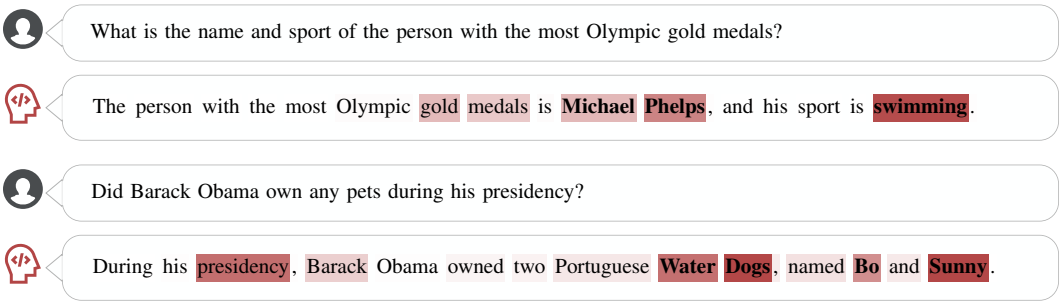


Figure 6: Learned ATWU scores on unseen real-entity examples. The examples use TOFU-style prompts with fictional entities replaced by real people and factual information.

B Unlearning Metrics

Evaluating an unlearned model requires measuring three orthogonal properties simultaneously: (i) how thoroughly targeted knowledge has been removed, (ii) how robust that removal is under mild perturbations of the prompt, and (iii) how much of the model's retain-set and general utility survives the intervention. No single score captures all three, so we report a panel of complementary metrics organized by purpose. Sections B.1 and B.2 describe the two primary evaluation protocols we inherit from prior work: a token-level extraction score probing verbatim memorization, and a judge-based family that stress-tests forgetting against paraphrase and in-context-relearning attacks. Section B.3 describes auxiliary utility probes that monitor general post-unlearning usability. For the RWKU benchmark we additionally employ the native ROUGE-L-based evaluation panel of Jin et al. [2024], described in Section B.4, as a cheap tuning surrogate during our Stage 1 hyperparameter search. WMDP is instead evaluated with the benchmark's own multiple-choice protocol rather than with the judge-based metrics (Section B.5). Finally, Section B.6 introduces three baseline-relative summary scores that normalize the preceding metrics against the original checkpoint so that runs are directly comparable across models and forget sets. These are the numbers we use to rank methods at a glance.

B.1 Token-level Extraction Metrics

Extraction Strength (ES)**.** Following Wang et al. [2025b] (and the TOFU evaluation of Maini et al. [2024]), we measure how strongly a response is parameterized in the model's weights using the *Extraction Strength*. For a question–answer pair (q, a) with answer tokens $\mathrm{a} = (\mathrm{a}_1, \ldots, \mathrm{a}_T)$, let $M(\cdot)$ denote greedy decoding. Define the shortest extractable prefix

$$k^\star(\mathrm{q}, \mathrm{a}) = \min \left\{ k \in \{0, 1, \ldots, T\} \ : \ M\big(\mathrm{q} \oplus \mathrm{a}_{1:k}\big) = \mathrm{a}_{k+1:T} \right\}. \tag{8}$$

Whenever no strict prefix $k < T$ reconstructs the suffix, the minimum collapses to $k^\star(\mathrm{q}, \mathrm{a}) = T$, corresponding to the worst case. The per-sample extraction strength is then

$$\mathrm{ES}(\mathrm{q}, \mathrm{a}) = 1 - \frac{k^\star(\mathrm{q}, \mathrm{a})}{T}, \qquad \mathrm{ES}(\mathrm{q}, \mathrm{a}) \in [0, 1]. \tag{9}$$

$\mathrm{ES} = 1$ indicates the answer is reproduced verbatim from the question alone (the model has fully memorized the response); $\mathrm{ES} = 0$ indicates even the full ground-truth answer prefix fails to elicit the remaining tokens. We report averages over the forget and retain splits,

$$\mathrm{ES}_{\mathcal{F}} = \frac{1}{|\mathcal{F}|} \sum_{(\mathrm{q},\mathrm{a}) \in \mathcal{F}} \mathrm{ES}(\mathrm{q}, \mathrm{a}), \qquad \mathrm{ES}_{\mathcal{R}} = \frac{1}{|\mathcal{R}|} \sum_{(\mathrm{q},\mathrm{a}) \in \mathcal{R}} \mathrm{ES}(\mathrm{q}, \mathrm{a}), \tag{10}$$

and summarize the unlearning/retention trade-off with

$$\mathrm{ES}_\Delta = \mathrm{ES}_{\mathcal{R}} - \mathrm{ES}_{\mathcal{F}}. \tag{11}$$

Motivation. Unlike likelihood-based scores, ES probes the actual generative behavior of the unlearned model, which is what an adversary would exploit. A method that merely suppresses the probability of the forget answer without disrupting its greedy decoding path is credited by perplexity-style metrics but penalized by $\mathrm{ES}_{\mathcal{F}}$. ES_Δ provides a single scalar capturing the gap between targeted removal and preserved recall, which we use as the objective during hyperparameter search.

Shortcomings. ES is defined through an *exact* suffix match, which makes it brittle to trivial deviations. Consider a generated response that agrees with $\mathrm{a} = (\mathrm{a}_1, \ldots, \mathrm{a}_T)$ on every token except the last: $M(\mathrm{q}) = (\mathrm{a}_1, \ldots, \mathrm{a}_{T-1}, x)$ with $x \neq \mathrm{a}_T$. Under greedy decoding the same mismatch persists for every teacher-forced prefix; for any $k < T$, $M(\mathrm{q} \oplus \mathrm{a}_{1:k})$ still deterministically produces $(\mathrm{a}_{k+1}, \ldots, \mathrm{a}_{T-1}, x)$, so $\mathrm{a}_{k+1:T}$ is never recovered and $k^\star(\mathrm{q}, \mathrm{a}) = T$, yielding $\mathrm{ES}(\mathrm{q}, \mathrm{a}) = 0$. Semantically the model has reproduced almost the entire answer verbatim, yet the score is indistinguishable from a response that disagrees from the very first token. The same pathology afflicts any mismatch in a single late position (e.g. a swapped named entity near the end of the answer), or cases where the model emits a stylistic token (e.g. punctuation, casing, whitespace) that differs from the reference. ES is therefore best read as a conservative *lower bound* on memorization: a low $\mathrm{ES}_{\mathcal{F}}$ does not certify forgetting, which is why we pair it with the judge-based scores of Section B.2.

B.2 Robustness-Aware Forget and Retain Quality

The remaining metrics in this section are adopted from the evaluation protocol of Singh et al. [2025], which replaces brittle ROUGE-style matching with an LLM-as-judge score $\mathcal{J}$ and stress-tests each unlearned model with paraphrased and in-context-primed queries. Given a query q and reference answer a, the judge returns $\mathcal{J}(\mathrm{q}, \mathrm{a}, \hat{\mathrm{a}}\ ;\ \mathrm{p}) \in \{0, 1\}$ grading a candidate response $\hat{\mathrm{a}}$; the scores below are averages or worst-cases of $\mathcal{J}$ under different query transformations. The symbol p denotes the fixed system prompt we supply to the judge LLM: it defines the grading rubric and the expected output format, and is held identical across all of $\mathcal{J}_{\mathrm{P}}, \mathcal{J}_{\mathrm{ICR}}, \mathcal{J}_{\mathrm{W}}$, and $\mathcal{J}_{\mathrm{AVG}}$ so that the resulting scores are commensurable. The exact rubric wording and format instructions we use, as well as the prompts used to generate paraphrases, are included in the supplementary material.

For any query q we generate N_P paraphrases $\{\mathrm{q}^{(i)}\}_{i=1}^{N_P}$ that preserve the semantic content of q, and denote the model's greedy response to each by $\hat{\mathrm{a}}^{(i)} = M(\mathrm{q}^{(i)})$. The four metrics below reuse this notation.

Paraphrase score ($\mathcal{J}_{\mathrm{P}}$). On the forget set we report the worst-case judge score across the N_P paraphrases,

$$\mathcal{J}_{\mathrm{P}}(\mathrm{q}, \mathrm{a}) \;=\; \max_{i \in [N_P]} \mathcal{J}\big(\mathrm{q}^{(i)}, \mathrm{a}, \hat{\mathrm{a}}^{(i)}\ ;\ \mathrm{p}\big), \qquad (\mathrm{q}, \mathrm{a}) \in \mathcal{F}. \tag{12}$$

A low $\mathcal{J}_{\mathrm{P}}$ certifies that *no* paraphrase of q recovered the forgotten content, i.e. forgetting generalizes beyond the exact training phrasing. We depart here from Singh et al. [2025], who define $\mathcal{J}_{\mathrm{P}}$ as the *mean* over paraphrases and do not use it further in their evaluation protocol. Defining it as the worst-case instead makes it directly useful as a building block of our final forget-set score.

In-context-relearning score ($\mathcal{J}_{\mathrm{ICR}}$). To probe whether forgotten information resurfaces when the model is primed with related content, for each paraphrase $\mathrm{q}^{(i)}$ we independently sample a small random subset $\mathrm{d}^{(i)} \subset \mathcal{R}$ of retain-set Q/A pairs and prepend them as in-context demonstrations. On RWKU, the demonstrations are matched to the probe type being scored (Section B.4): QA demonstrations for QA probes, FB demonstrations for FB probes, and AA demonstrations for AA probes. Let $\tilde{\mathrm{a}}^{(i)} = M(\mathrm{d}^{(i)} \oplus \mathrm{q}^{(i)})$ denote the resulting generation. The ICR score is the worst case across paraphrases,

$$\mathcal{J}_{\mathrm{ICR}}(\mathrm{q}, \mathrm{a}) \;=\; \max_{i \in [N_P]} \mathcal{J}\big(\mathrm{q}^{(i)}, \mathrm{a}, \tilde{\mathrm{a}}^{(i)}\ ;\ \mathrm{p}\big). \tag{13}$$

Structurally, $\mathcal{J}_{\mathrm{ICR}}$ mirrors $\mathcal{J}_{\mathrm{P}}$, with both taking a sample-wise $\max$ over the same paraphrase set $\{\mathrm{q}^{(i)}\}_{i=1}^{N_P}$. The two differ only in the response being scored; $\mathcal{J}_{\mathrm{ICR}}$ uses the primed generation $\tilde{\mathrm{a}}^{(i)}$, whereas $\mathcal{J}_{\mathrm{P}}$ uses the zero-shot response $\hat{\mathrm{a}}^{(i)}$. A genuinely forgotten fact should remain unrecoverable even after priming, so $\mathcal{J}_{\mathrm{ICR}}$ captures residual memorization that plain paraphrase attacks alone would miss.

Worst-case forget quality ($\mathcal{J}_{\mathrm{W}}$). Since $\mathcal{J}_{\mathrm{P}}$ and $\mathcal{J}_{\mathrm{ICR}}$ are already sample-wise worst cases, the primary robustness score reduces to their maximum,

$$\mathcal{J}_{\mathrm{W}}(\mathrm{q}, \mathrm{a}) \;=\; \max\big(\mathcal{J}_{\mathrm{P}}(\mathrm{q}, \mathrm{a}),\ \mathcal{J}_{\mathrm{ICR}}(\mathrm{q}, \mathrm{a})\big). \tag{14}$$

This is the main metric we track on $\mathcal{F}$: a low $\mathcal{J}_{\mathrm{W}}$ certifies that *no* lightweight attack among those considered elicited the forgotten content.

Average retain quality ($\mathcal{J}_{\mathrm{AVG}}$). On the retain set, worst-case evaluation is overly pessimistic because paraphrases can legitimately push the model toward equally valid rephrasings of the answer. We therefore report the mean judge score across the N_P paraphrases,

$$\mathcal{J}_{\mathrm{AVG}}(\mathrm{q}, \mathrm{a}) \;=\; \frac{1}{N_P} \sum_{i=1}^{N_P} \mathcal{J}\big(\mathrm{q}^{(i)}, \mathrm{a}, \hat{\mathrm{a}}^{(i)}\ ;\ \mathrm{p}\big), \qquad (\mathrm{q}, \mathrm{a}) \in \mathcal{R}. \tag{15}$$

Note that the functional form matches $\mathcal{J}_{\mathrm{P}}$; the distinction is the split on which the score is aggregated. A high $\mathcal{J}_{\mathrm{AVG}}$ indicates that the model still reliably answers questions it was meant to keep.

B.3 Utility Metrics

Beyond the targeted forget/retain splits we monitor three signals of *general* post-unlearning usability, again following Singh et al. [2025]. We treat these signals as *utility-preservation probes* rather than optimization targets: the relevant failure mode is degradation relative to the original checkpoint, namely a drop in MMLU, a drop in Rep., or WR $\ll 50.0$. Conversely, parity or occasional improvements are not penalized, since they may reflect benign side effects of the unlearning procedure or measurement noise rather than harmful changes.

MMLU. We evaluate multiple-choice accuracy on MMLU [Hendrycks et al., 2021] using argmax selection over the four options rather than open-ended generation, computed with the `lm-evaluation-harness` of Gao et al. [2024]. MMLU tracks the broad world knowledge of the model, which collateral forgetting can silently erode. We therefore interpret MMLU asymmetrically: scores at or above the original checkpoint are taken to preserve this axis of utility.

Repetitiveness (Rep.). Aggressive unlearning loss functions often cause the model to collapse into degenerate, low-entropy outputs. Following the *Fluency* metric of Jin et al. [2024], we measure repetitiveness from the bi- and tri-gram frequencies of generations produced on the AlpacaEval instruction set. Lower values indicate more frequently repeated n-grams, and thus flag pathological generation behavior that task metrics alone would not reveal. We likewise interpret Rep. asymmetrically, flagging only meaningful drops relative to the original checkpoint as degradation.

Win Rate (WR)**.** Repetitiveness alone does not capture whether responses are actually useful, so we additionally compare the instruction-following quality of the unlearned model against a *reference model* using an LLM-as-judge in the style of AlpacaEval. On TOFU, the reference is the retain model of the corresponding split, i.e., the model fine-tuned only on the retain set (e.g., `retain90` for `forget10`), which represents the behavior an ideal unlearning procedure should recover. On RWKU and WMDP, where no such model is available, the reference is the original checkpoint. For each prompt the judge is shown the question together with both responses in a single call and is asked to rate each on a 1–10 scale along with a short justification; wins, losses, and ties are then determined by comparing the two scores. For every prompt we record whether the unlearned model received a higher score (U_{Wins}), lower score (U_{Losses}), or a tie (U_{Ties}) relative to the reference model, and compute

$$\text{WR} \;=\; 100 \cdot \frac{U_{\text{Wins}} + 0.5\, U_{\text{Ties}}}{U_{\text{Wins}} + U_{\text{Losses}} + U_{\text{Ties}}}. \tag{16}$$

We note that the judge is not informed which response originates from the unlearned model and which from the reference: the prompt labels them only as "Assistant 1" and "Assistant 2", and nowhere in the system prompt, user prompt, or scoring criteria do we reference unlearning, forgetting, or model identity. Consequently the judge has no semantic cue that could bias it toward either model. The exact judge prompt used for this evaluation is included in the supplementary material. By construction, the reference model has WR $= 50$ against itself. WR $\ll 50$ is the clear failure mode: it indicates that unlearning has degraded general response quality relative to the reference model. WR ≈ 50 is the expected and desired outcome, signaling that quality has been preserved. We depart here from Singh et al. [2025], who argue that WR should be *as close as possible* to 50 and treat deviations above 50 as undesirable. In our setting, we do not penalize WR > 50: such values may reflect genuine quality improvements, regularization side-effects of the unlearning objective, or judge noise, none of which constitutes the utility-degradation failure mode we aim to detect.

B.4 Native RWKU Benchmark Metrics

Alongside the benchmark, Jin et al. [2024] ship an evaluation panel that we use for two purposes in this paper: as the tuning surrogate for the RWKU Stage 1 search (Section C.3), and as the basis of the detailed per-probe RWKU tables reported later in the appendix. Responses are not collected on a single probe set but across three complementary probe types, applied both to the forget targets and to a separately-curated set of adjacent facts that must be preserved.

The first probe type, **FB**, consists of fill-in-the-blank cloze items that test whether the target knowledge can still be elicited from a partial prompt. The second, **QA**, is open-form question answering, where the model is asked directly about the target fact. The third, **AA**, bundles a family of adversarial

reformulations of the same queries (prefix injection, role-playing, reverse querying, affirmative suffixes, cross-lingual variants, and several other jailbreak-style attacks) and reports the worst-case response across them. Each probe's response is scored by the ROUGE-L recall against the reference answer, and the aggregate **All** is the mean over the probe types applied to that split.

Probes are applied to two disjoint splits of the benchmark. The **Forget Set** ($\mathcal{F}$) contains probes directed at the targeted knowledge across all three probe types FB, QA, and AA; lower ROUGE-L (↓) indicates that the model no longer reproduces the forgotten content. The **Neighbor Set** ($\mathcal{N}$) contains probes directed at related but non-targeted knowledge that the benchmark designates as *must-preserve*; following the benchmark, only the FB and QA probe types are reported on the Neighbor Set, since adversarial attacks are a forget-side stress test rather than a utility measurement. Higher ROUGE-L (↑) indicates that retain-adjacent facts survive the intervention. The Neighbor Set therefore plays on RWKU the role that the retain split $\mathcal{R}$ plays on TOFU.

For hyperparameter tuning we summarize the two panels with a single scalar,

$$N_\Delta \;=\; \text{ROUGE-L}_{\mathcal{N},\text{All}} \;-\; \text{ROUGE-L}_{\mathcal{F},\text{All}}, \tag{17}$$

the signed gap between the two "All" aggregates. Higher N_Δ means better forgetting without collateral damage to neighboring knowledge, and we use it as the selection criterion at Stage 1 on RWKU.

We emphasize that the ROUGE-L-based scores above are a weaker signal than the judge-based metrics $\mathcal{J}_\text{P}, \mathcal{J}_\text{ICR}, \mathcal{J}_\text{W}, \mathcal{J}_\text{AVG}$ of Section B.2: as discussed by Singh et al. [2025], ROUGE penalizes semantically correct responses that are phrased differently from the reference and conversely credits generic or repetitive outputs that happen to share surface n-grams, so it systematically mislabels unlearning success and failure and can reorder methods with respect to a judge. We therefore treat these native RWKU scores as a legacy reporting panel, useful as a cheap tuning surrogate and for comparability with prior work that reports them, but consider the judge-based metrics our primary evaluation on RWKU.

Judge-based scores on RWKU. The judge-based scores of Section B.2 are computed on the same probes. On the forget side, $\mathcal{J}_\text{P}$, $\mathcal{J}_\text{ICR}$, and hence $\mathcal{J}_\text{W}$ are pooled over the FB, QA, and AA probes of the forget set, so the FQ and UQ values we report on RWKU aggregate all three probe types, while Table 13 reports $\mathcal{J}_\text{W}$ separately for each forget probe type.

B.5 WMDP Metrics

WMDP [Li et al., 2024] measures hazardous knowledge with 3,668 four-option multiple-choice questions in biosecurity, cybersecurity, and chemistry. Following the RMU experiments of Li et al. [2024], we target only the biosecurity and cybersecurity subsets, WMDP-Bio (1,273 questions) and WMDP-Cyber (1,987 questions).

Forget metrics. We report zero-shot multiple-choice accuracy on each subset (Bio and Cyber), computed with the `wmdp_bio` and `wmdp_cyber` tasks of the `lm-evaluation-harness` [Gao et al., 2024], together with their unweighted mean,

$$\text{WMDP} \;=\; \tfrac{1}{2}\,\big(\text{Bio} + \text{Cyber}\big), \tag{18}$$

so that both domains count equally regardless of their number of questions. Lower accuracy indicates more forgetting, with random-guessing being 25%: accuracy below chance does not indicate stronger forgetting. WMDP provides neither free-form reference answers for a judge to compare against nor a retain question set, so the scores of Sections B.2 and B.6 are not defined on it. The multiple-choice protocol is the standard WMDP evaluation, and its forget-side scores require no judge.

Utility metrics. We report MMLU, Rep., and WR exactly as defined in Section B.3, and flag utility degradation with the same thresholds as in the main paper: an MMLU drop of more than 2 percentage points, a Rep. drop of more than 6%, or $\text{WR} < 45$.

B.6 Baseline-Relative Summary Metrics

The raw scores $\mathcal{J}_\text{W}$ and $\mathcal{J}_\text{AVG}$ are not directly comparable across models or forget splits, because the *operating point* of the original checkpoint varies: a final $\mathcal{J}_\text{W} = 0.2$ is excellent when the original

model scored 0.9 on the same forget set, but mediocre when the original already sat at 0.4. We therefore summarize each run by three baseline-relative scores, reported as *fractional* drops with respect to the original checkpoint. Writing $\mathcal{J}_{\mathrm{W}}^{\mathrm{orig}}$ and $\mathcal{J}_{\mathrm{AVG}}^{\mathrm{orig}}$ for the original model's scores on the same splits, and using $[x]_+ = \max(0, x)$ throughout, we clip each drop below at zero so that accidental increases (typically judge noise) are not miscounted as progress. This normalization puts all three metrics on the same $[0, 1]$ scale regardless of the original operating point, and for readability we report them as percentages in $[0, 100]$ throughout the paper. These summary scores are used on TOFU and RWKU, while WMDP uses the metrics of Section B.5.

Forget Quality (FQ). FQ measures the fractional reduction in the worst-case forget-set judge score attributable to unlearning,

$$\mathrm{FQ} \;=\; \left[\, 1 - \frac{\mathcal{J}_{\mathrm{W}}}{\mathcal{J}_{\mathrm{W}}^{\mathrm{orig}}} \,\right]_+ . \tag{19}$$

It ranges from 0 (no forgetting achieved, $\mathcal{J}_{\mathrm{W}} \geq \mathcal{J}_{\mathrm{W}}^{\mathrm{orig}}$) up to its ideal value of 1, reached only by a method that drives the worst-case judge score on the forget set to zero.

Retain Degradation (RD). RD measures the collateral cost of unlearning as the fractional loss in the average retain-set judge score,

$$\mathrm{RD} \;=\; \left[\, 1 - \frac{\mathcal{J}_{\mathrm{AVG}}}{\mathcal{J}_{\mathrm{AVG}}^{\mathrm{orig}}} \,\right]_+ . \tag{20}$$

The ideal value is $\mathrm{RD} = 0$, indicating that the retain split is answered as well as before unlearning; $\mathrm{RD} = 1$ would correspond to a complete collapse of retain-side quality.

Unlearning Quality (UQ). UQ combines the two by subtracting retain-side damage from forget-side gain,

$$\mathrm{UQ} \;=\; \left[\, \mathrm{FQ} - \mathrm{RD} \,\right]_+ . \tag{21}$$

It is the fraction of forget-quality improvement that was *not* paid for by retain-set degradation, and lives on the same $[0, 1]$ scale with ideal value 1. A value of $\mathrm{UQ} = 0$ flags methods whose retain damage cancels out any forgetting progress, regardless of how low $\mathcal{J}_{\mathrm{W}}$ is in isolation.

Motivation. The triple captures the two competing goals of unlearning – removing targeted knowledge and preserving the rest – together with their net trade-off. FQ and RD make each axis legible on its own, while UQ collapses the trade-off into a single scalar that we use to rank methods in our main tables, since a method that pushes $\mathcal{J}_{\mathrm{W}}$ down by uniformly blunting the model should not be credited with genuine unlearning.

C Experimental Details

This section details the setup used for the experiments reported in the main paper. We describe the benchmarks and base models (Section C.1), the generated corpora used for WMDP unlearning (Section C.2), the hyperparameter-tuning protocol (Section C.3), the method-specific hyperparameter search ranges and selected configurations (Section C.4), and the final unlearning runs and evaluation protocol (Section C.5). Finally, we report the computational cost of ATWU (Section C.6) and examine the sensitivity of our judge-based results to the choice of judge model and rubric (Section C.7).

C.1 Benchmarks and Base Models

We evaluate on three LLM-unlearning benchmarks.

TOFU [Maini et al., 2024]. A synthetic question-answering benchmark in which the base model has been fine-tuned on a corpus of fictitious author biographies. We use the three standard forget splits supplied with the benchmark, `forget01`, `forget05`, and `forget10`, covering progressively larger fractions of the fine-tuning corpus. We consider two base model sizes, `Llama-3.2-1B-Instruct`[4] and `Llama-3.1-8B-Instruct` [Grattafiori et al., 2024], and use the TOFU-fine-tuned checkpoints released by the `open-unlearning` project [Dorna et al., 2025],[5] whose codebase[6] our TOFU experiments extend in two ways: we add paraphrase-based evaluation, and integrate JensUn alongside our proposed methods.

RWKU [Jin et al., 2024]. A real-world knowledge-unlearning benchmark that targets knowledge about public figures present in a pretrained LLM. We evaluate on the 3.8B-parameter `Phi-3-Mini-4k-Instruct` [Abdin et al., 2024]. Following Singh et al. [2025], we perform *10-target batch unlearning*: rather than unlearning one target at a time, a single unlearning run jointly removes knowledge about ten public figures. We fix one specific choice of ten subjects and refer to it as our *canonical ten-subject RWKU batch* throughout the paper. Since RWKU does not explicitly provide retain data, during unlearning we use a *disjoint* set of ten other subjects as a surrogate retain set. We adapt the official RWKU codebase[7] in two ways: we add retain-set support, which the upstream release lacks since it only supports forget-only methods, and integrate GradDiff, SimNPO, JensUn, WGA, SatImp, alongside our proposed methods.

WMDP [Li et al., 2024]. A multiple-choice benchmark of hazardous knowledge in biosecurity, cybersecurity, and chemistry. We target the biosecurity and cybersecurity subsets and evaluate as described in Section B.5. The base model is `Zephyr-7B-beta` [Tunstall et al., 2023],[8] the model used in the RMU experiments of Li et al. [2024]. Unlearning uses the generated forget and retain corpora described in Section C.2. We compare GradDiff, NPO, SimNPO, JensUn, RMU, WGA, SatImp, and ATWU. DPO is excluded because it requires a preferred and a rejected response for every forget example, which a corpus of passages does not provide. Our WMDP code[9] extends the same `open-unlearning` codebase with a data loader that feeds each optimizer step from a single forget domain, alternating between the two as in the official RMU implementation.

C.2 WMDP Unlearning Corpora

Released corpora and document selection. We distinguish the WMDP evaluation benchmark from the corpora used to perform unlearning. WMDP contains 3,668 multiple-choice questions, of which we use the 1,273 WMDP-Bio and 1,987 WMDP-Cyber questions, while its released forget corpora contain approximately 24,500 Bio documents and 1,000 Cyber documents [Li et al., 2024]. However, the benchmark does not define a canonical, method-independent training subset for a fixed unlearning budget. For example, the official RMU reproduction processes the corpora without shuffling and alternates between Bio and Cyber for 150 updates with batch size 4.[10] It therefore

[4] https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct
[5] https://huggingface.co/collections/open-unlearning/tofu-new-models.
[6] https://github.com/locuslab/open-unlearning
[7] https://github.com/jinzhuoran/RWKU
[8] https://huggingface.co/HuggingFaceH4/zephyr-7b-beta
[9] https://github.com/tml-epfl/ATWU/tree/wmdp
[10] This behavior follows directly from the official data loader, training loop, and Zephyr configuration.

uses the first 300 documents of each corpus that pass the loader's 50-character minimum-length filter: approximately 1.2% of Bio but 30% of Cyber. Although deterministic, this selection is tied to dataset order rather than semantic coverage; implementations that shuffle or sample the corpora may consequently train on substantially different documents.

Document selection as a source of variance. In preliminary experiments where the consumed subset varied across seeds, we observed significant variation across **all** unlearning methods, with the effect being most pronounced on WMDP-Bio. This suggests that the released documents are heterogeneous in their relevance to the knowledge measured by WMDP and that document selection can become a hidden source of experimental variance. We therefore construct a fixed, balanced corpus so that every method operates on the same domain coverage and corpus size.

Model-elicited forget corpora. Our construction follows the model-elicitation principle used by the RWKU unlearning protocol. Because the original training documents associated with an RWKU target are unavailable, RWKU prompts each target model to generate biographies and questions about that target and uses these model-specific outputs as unlearning data [Jin et al., 2024]. We apply the same principle at the domain level: `Zephyr-7B-beta` [Tunstall et al., 2023] is prompted to verbalize its knowledge of public WMDP topic areas, and these outputs form its forget corpus. This ensures that the unlearning objective operates on knowledge the target checkpoint can express, rather than on arbitrary external documents that the model may not have internalized. Unlike RWKU, we freeze the resulting corpus and use it for every method and seed, eliminating corpus identity as a confounding variable.

Generation procedure. Corpus generation proceeds in two stages. First, `Zephyr` expands 12 high-level categories per domain, which we wrote to follow the topic areas described by WMDP and release with the code, into up to 50 concrete technical subtopics per category (600 for Bio and 596 for Cyber). We sample this expansion once and freeze the resulting subtopic lists. Second, `Zephyr` generates long-form technical passages for these subtopics using eight expository framings. Subtopics are traversed round-robin across categories so that stopping generation at the desired corpus size preserves broad and approximately balanced coverage. Generation targets 3,200 passages per domain; because the count is checked after each batch, it stops slightly above the target, yielding 3,215 Bio and 3,215 Cyber passages. Generation uses seed 0, temperature 0.9, top-p 0.95, a repetition penalty of 1.05, batch size 48, and at most 820 (Bio) or 560 (Cyber) new tokens per passage. We remove boilerplate openings and reject refusals (detected in the first 600 characters), passages shorter than 250 words, strongly repetitive generations (more than 15% repeated 8-grams), and outputs with more than 5% non-ASCII characters. The prompts contain only high-level topic descriptions: no WMDP question, answer, or answer choice is used during either generation stage.

Retain corpus. The retain corpus is not labeled supervision or a source from which the model must learn new facts. Instead, it provides contexts on which the unlearned model is encouraged to preserve the original model's behavior. It therefore need not reproduce the original pretraining corpus, but should broadly cover the capabilities that the unlearning procedure is intended to preserve. With the same procedure and at most 560 new tokens per passage, we generate 4,029 retain passages (target 4,000) spanning all 57 MMLU subjects (14 subtopics per subject), since MMLU is also our principal utility measure. This aligns the retain distribution with the capabilities evaluated after unlearning. Generating the forget and retain passages with the same model and in the same expository format also controls for superficial source and style differences, preventing methods from distinguishing the two sets merely through formatting artifacts. We deliberately retain neighboring subjects such as general biology, virology, and computer security: the desired behavior is selective removal of hazardous knowledge rather than indiscriminate suppression of entire scientific or technical domains.

How runs consume the corpora. Each WMDP run takes 150 optimizer steps with an effective batch size of 32, and every step draws from a single forget domain, alternating between Cyber and Bio (Section C.5). A run therefore consumes 2,400 forget passages per domain, and the remaining 815 passages per domain are never used for training. The consumed forget passages are the first 2,400 entries of a fixed random permutation of each corpus and are identical for every method and seed: the seed changes only their order and which retain passages are drawn (without replacement) to accompany them. Passages are truncated to their first 768 tokens for Bio and 512 tokens for Cyber and retain, following the RMU implementation.

Preliminary and final corpora. Hyperparameters for all methods were selected (Section C.3) on a preliminary corpus produced by an earlier, non-deterministic run of the same generation pipeline. This preliminary corpus is not released and cannot be recreated exactly. The final results retrain the selected configurations, without any retuning, on the final, reproducible corpus described above, using seeds 1, 2, and 3.

Scope and reproducibility. The generated corpus is not intended as a replacement for WMDP or as an authoritative reference corpus. It is controlled optimization data designed to make method comparisons less sensitive to arbitrary document selection; evaluation remains unchanged on the standard WMDP multiple-choice questions. We do not distribute the generated passages themselves, since they contain long-form material on biological and cybersecurity hazards. Instead, we release the generation code together with the prompts and frozen subtopics, and pin the `Zephyr` checkpoint revision, sampling parameters, batch size, and random seed. Together, these choices make the corpus reproducible within the same software and hardware environment.

C.3 Hyperparameter Tuning Protocol

Each unlearning method exposes a set of loss-specific hyperparameters (learning rate, loss-weighting coefficients, optional auxiliary objectives, etc.). We tune these per method and per benchmark using Bayesian optimization via `optuna` [Akiba et al., 2019], maximizing a cheap-to-compute surrogate of unlearning quality rather than the paraphrase- and judge-based evaluation of Sections B.2 and B.3 that we actually *report*: running the full pipeline at every trial is infeasible at our compute and API budget, since the ablations, and a single final-evaluation pass already incurs roughly $180 in judge-LLM API costs. We therefore run the judge-based evaluation only on the final selected checkpoints, validating the surrogate after the fact. Throughout tuning, the optimizer, batch size, gradient-accumulation steps, precision, and learning-rate schedule are held fixed at the final-run values of Section C.5; only the method-specific hyperparameters listed in the per-stage descriptions below are searched. On TOFU and RWKU, tuning proceeds in two stages; WMDP uses the single-stage protocol described at the end of this subsection.

Stage 1: coarse search. For each method we run an initial search over all method-specific hyperparameters on the largest forget split of the benchmark (`forget10` for TOFU, our canonical ten-subject batch for RWKU). On TOFU this stage runs on the small base model, `Llama-3.2-1B-Instruct`, for 50 trials at the full training budget of Section C.5, with the first trial seeded at the hyperparameters recommended by each method's original paper (for ATWU, see Section C.4); trials are selected by $\mathrm{ES}_\Delta = \mathrm{ES}_\mathcal{R} - \mathrm{ES}_\mathcal{F}$ (Section B.1). On RWKU this stage runs on `Phi-3-Mini-4k-Instruct` for 20 trials of a single epoch at a fixed learning rate, optimizing only the non-LR hyperparameters and selecting by the ROUGE-L-based N_Δ surrogate defined in Section B.4. The first trial is seeded with each method's final TOFU 8B configuration—the one used for the corresponding final unlearning runs in Section C.5—under the assumption that well-tuned settings transfer across benchmarks. For most methods, we keep the learning rate unchanged in this pass. For GradDiff and JensUn, however, we reduce it by a factor of 10; the same reduction is also applied to WGA and SatImp as variants of GradDiff. This is informed by Singh et al. [2025], who report that these methods require markedly smaller learning rates on RWKU than on a QA dataset to produce stable unlearning, so anchoring at the TOFU 8B value directly would overshoot. Finally, the motivation for holding the learning rate fixed at all in this RWKU pass is twofold: a single-epoch budget gives an unreliable signal about the correct learning rate, so we defer LR tuning to Stage 2 where the full epoch budget applies, and reducing the number of jointly tuned parameters yields tighter estimates of the remaining method-specific hyperparameters within the same search budget.

Stage 2: learning-rate transfer. In both benchmarks we transfer the Stage 1 configuration to the evaluation setting and tune only the learning rate. We run 5 trials per method, sweeping the learning rate on a log-scale grid and holding all other hyperparameters at their Stage 1 values. On TOFU the transfer is across model scale: the Stage 1 configuration on the 1B model is ported to `Llama-3.1-8B-Instruct` and the learning rate is retuned, since re-running the full search at 8B would be prohibitive. On RWKU the transfer is across training budget: the Stage 1 configuration (tuned at one epoch) is run at the final-run epoch count so that the learning rate is selected against the full schedule, still on `Phi-3-Mini-4k-Instruct`. Selection is automatic on both benchmarks; we

	lr	R-L$_{\mathcal{F}}$ ↓	R-L$_{\mathcal{N}}$ ↑	N_Δ ↑		lr	R-L$_{\mathcal{F}}$ ↓	R-L$_{\mathcal{N}}$ ↑	N_Δ ↑
DPO	1.18×10^{-5}	0.499	0.590	0.090	GradDiff	1.18×10^{-6}	0.053	0.445	0.392
	2.70×10^{-5}	0.435	0.450	0.015		1.90×10^{-6}	0.045	0.399	0.354
	1.98×10^{-5}	0.491	0.501	0.010		2.70×10^{-6}	0.033	0.346	0.313
NPO	2.60×10^{-5}	0.236	0.541	0.305	WGA	1.54×10^{-6}	0.061	0.515	0.454
	1.18×10^{-5}	0.325	0.597	0.272		1.57×10^{-6}	0.063	0.502	0.438
	1.98×10^{-5}	0.257	0.509	0.252		3.00×10^{-6}	0.034	0.427	0.393
SimNPO	2.08×10^{-5}	0.218	0.561	0.343	SatImp	1.98×10^{-6}	0.132	0.617	0.485
	1.18×10^{-5}	0.254	0.594	0.341		4.48×10^{-6}	0.069	0.533	0.464
	2.70×10^{-5}	0.218	0.541	0.323		3.00×10^{-6}	0.112	0.566	0.454
JensUn	1.54×10^{-6}	0.062	0.434	0.372	ATWU	1.83×10^{-6}	0.050	0.562	0.512
	3.00×10^{-6}	0.040	0.408	0.368		1.74×10^{-6}	0.072	0.584	0.511
	3.98×10^{-6}	0.038	0.377	0.339		1.37×10^{-6}	0.100	0.610	0.510

Table 5: Top-3 Stage 2 learning-rate trials per method on RWKU, ranked by N_Δ. Green shading marks the trial used for the final unlearning run: the top-N_Δ trial throughout, except for SatImp, whose top trial was discarded for inadequate forgetting in favor of the second-best.

use ES$_\Delta$ on TOFU and N_Δ on RWKU. The only exception is SatImp on RWKU, where the top-N_Δ trial had inadequate forget performance and we therefore selected the second-best trial (see Table 5).

WMDP: single-stage search with seed validation. On WMDP, both the search and the seed validation below were run on the preliminary corpus of Section C.2. For each method, we run 20 `optuna` trials with the TPE sampler (sampler seed 0), each at the full 150-step training budget of Section C.5 and with training seed 0. Trial 0 is a warm start from preliminary sweeps, which were also used to set the search ranges, and trials 1–4 are sampled at random before TPE takes over. Trials maximize MMLU minus the average above-chance WMDP accuracy,

$$\text{score} \;=\; \text{MMLU} - \tfrac{1}{2}\big([\text{Bio}-25]_+ + [\text{Cyber}-25]_+\big),$$

with all accuracies in percent. Flooring each subset at chance prevents below-chance accuracy on one domain from compensating for insufficient forgetting on the other. The three trials with the highest score are then rerun with seeds 1 and 2, and we select the one whose worst seed is furthest inside both thresholds (MMLU $\geq 57\%$, WMDP $\leq 27\%$), i.e., the largest worst-seed margin $\min_s \min(\text{MMLU}_s - 57,\, 27 - \text{WMDP}_s)$. The selected configuration is then frozen. On the preliminary corpus, only ATWU and WGA met both thresholds on every seed, with worst-seed margins of $+0.34$ and $+0.21$ percentage points, respectively; all other methods had negative margins. Rep. and WR are not used for selection.

C.4 Hyperparameter Search Ranges and Results

This subsection specifies the concrete search ranges used in the two-stage tuning procedure of Section C.3 and reports the selected configurations for TOFU (Table 6), RWKU (Table 7), and WMDP (Table 8). Unless stated otherwise, all ranges are searched on a logarithmic scale. The coefficients α and γ denote the retain-loss and forget-loss weights, respectively, and are shared across methods. For most baselines, the method-specific grids follow a slightly extended version of those used by Yang et al. [2025]. On TOFU and RWKU, for our score-based methods, we fix $\lambda_{\text{ent}} = \lambda_{\text{w}} = 1$ and set $\lambda_{\text{pop}} = 15$ throughout, with budget $\rho = 0.2$ on TOFU and $\rho = 0.5$ on RWKU. An ablation on the different regularizers is presented in Table 20 and discussed in Section D.5. Throughout this subsection, hyperparameter values returned by `optuna` are rounded to two significant digits and performance scores to three decimal places for readability. The exact hyperparameter values used in the final runs reported in Tables 6 and 7 are provided in the supplementary material.

Stage 1: coarse search. For **both** datasets, we tune all non-fixed method-specific hyperparameters. All baselines, and ATWU on RWKU, search over $\alpha, \gamma \in [0.1, 5]$, and for TOFU, the baselines also

search $\mathrm{lr} \in [1 \times 10^{-6}, 5 \times 10^{-5}]$.[11] The per-method parameters tuned in this stage are as follows: for GradDiff, we try both `NLL` and `KL` as the retain loss;[12] for DPO and NPO, we search the inverse temperature β in the ranges $[0.2, 0.5]$ and $[0.05, 0.2]$, respectively; for SimNPO, we search the temperature β and reward margin δ in the ranges $[2.0, 3.0]$ and $[0.0, 2.0]$, respectively (the latter on a linear scale); for WGA, we tune the weight exponent β in the range $[0.1, 5]$; for SatImp, we search the saturation and importance exponents β_1 and β_2 in the ranges $[1.0, 10.0]$ and $[0.1, 5]$, respectively; for ETW, we search the temperature $\tau \in [0.1, 4.0]$ and use $\gamma \in [0.01, 2.0]$ instead; for RMU, we search the steering coefficient $c \in [0.5, 10.0]$, and the unlearning layer $\ell \in \{4, \ldots, 12\}$ (with the trainable parameters being the `mlp.down_proj` weights of layers $\{\ell - 2, \ell - 1, \ell\}$, following the WMDP RMU recipe); and for ATWU on RWKU, we tune $\beta \in [1.0, 10.0]$ and the scorer learning rate `slr` $\in [0.01, 0.5]$.

On TOFU, ATWU has no recommended configuration from prior work, so we set its search ranges from preliminary runs, analogous to the narrowed JensUn learning-rate range above: $\mathrm{lr} \in [1 \times 10^{-5}, 5 \times 10^{-5}]$, $\alpha \in [0.2, 2]$, and $\gamma \in [0.5, 8]$ on a logarithmic scale, and $\beta \in [3, 10]$ on a linear scale. In place of a single published default, we seed the search with ten configurations from these preliminary runs and draw the remaining 40 trials with TPE, keeping the same 50-trial budget as for the baselines. The selected configuration (Table 6) is one of these seed configurations and also lies within the ranges searched for the baselines. The scorer learning rate (0.05) and the regularization weights ($\lambda_{\mathrm{ent}} = \lambda_{\mathrm{w}} = 1$, $\lambda_{\mathrm{pop}} = 15$, $\rho = 0.2$) are fixed rather than tuned, so on TOFU ATWU tunes four hyperparameters: as many as DPO, NPO, WGA, and ETW, and fewer than SimNPO, SatImp, and RMU.

Stage 2: learning-rate transfer. On TOFU, the per-method range is anchored at the best Stage 1 learning rate: writing $\mathrm{lr}_m^\star$ for the learning rate selected for method m through the coarse search, we consider the range $\mathrm{lr} \in [\mathrm{lr}_m^\star/5,\ 5 \cdot \mathrm{lr}_m^\star]$, with the first trial seeded at $\mathrm{lr}_m^\star$ itself. For each method, the selected $\mathrm{lr}_m^\star$ are: 2.27×10^{-5} for DPO; 2.60×10^{-5} for NPO; 2.08×10^{-5} for SimNPO; 3.98×10^{-5} for JensUn; 1.90×10^{-5} for GradDiff; 1.57×10^{-5} for WGA; 1.98×10^{-5} for SatImp; 2.58×10^{-5} for ETW; 4.88×10^{-5} for RMU; and 2.00×10^{-5} for ATWU. On RWKU no analogous Stage 1 optimum is available, since the learning rate was held fixed throughout Stage 1; we instead choose method-specific log-scale ranges based on preliminary runs at the full Stage 2 training budget. The resulting ranges per method are: for DPO, NPO, and SimNPO, $[1 \times 10^{-5},\ 5 \times 10^{-5}]$; for GradDiff, JensUn, WGA, SatImp, and ATWU, $[1 \times 10^{-6},\ 1 \times 10^{-5}]$.

WMDP. The search ranges and warm-start configurations of every method on WMDP are available in the released code,[13] and the selected configurations are reported in Table 8. Unlike on TOFU and RWKU, the ATWU search on WMDP also tunes the budget and regularization weights, and we settle on $\rho = 0.40$, $\lambda_{\mathrm{pop}} = 10$, and $\lambda_{\mathrm{ent}} = \lambda_{\mathrm{w}} = 1$.

C.5 Final Unlearning Runs and Evaluation

Once a hyperparameter configuration is fixed for each (benchmark, base model, method) triple, we run the final unlearning procedure and evaluate the resulting checkpoints. On TOFU we run the procedure on each of the three forget splits `forget01`, `forget05`, and `forget10`; on RWKU we run it on our canonical ten-subject batch.

On TOFU, each run uses a single `NVIDIA A100-SXM4-80GB` GPU, trains for 10 epochs at batch size 8 with 4 gradient-accumulation steps (effective batch size 32), and follows the `open-unlearning` repository defaults [Dorna et al., 2025]: the `paged_adamw_32bit` optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 0.01, a one-epoch linear warmup followed by the default `linear` decay, gradient clipping at 1.0, and `bfloat16` mixed precision. On RWKU, each run uses two `NVIDIA A100-SXM4-80GB` GPUs with *model parallelism* only—the base model is sharded across the two GPUs while the batch is not replicated, so the per-step batch size is 8 and, with 4 gradient-accumulation steps, the effective batch size is 32—trains for 5 epochs, and follows the official RWKU

[11] For JensUn on TOFU, preliminary runs showed consistently weak forgetting at the smallest learning rates. We therefore narrowed its learning-rate search to $[5 \times 10^{-6}, 5 \times 10^{-5}]$, allowing a finer search in the more promising region.

[12] We only search over the retain loss on TOFU. On RWKU we fix it to `NLL`, since the TOFU Stage 1 search selected `NLL` consistently over `KL`, and we prefer to reduce tuning over parameters for which the choice already appears settled.

[13] https://github.com/tml-epfl/ATWU/tree/wmdp

Method	Shared		Method-Specific		Learning Rate	
	α	γ	β / β_1 / τ / c	δ / β_2 / slr / ℓ	`1B`	`8B`
GradDiff	0.80	0.12	—	—	1.90×10^{-5}	1.90×10^{-5}
DPO	0.15	3.80	0.21	—	2.27×10^{-5}	3.12×10^{-5}
NPO	4.10	0.12	0.10	—	2.60×10^{-5}	2.60×10^{-5}
SimNPO	1.28	1.49	2.82	0.03	2.08×10^{-5}	2.86×10^{-5}
JensUn	0.79	0.82	—	—	3.98×10^{-5}	2.66×10^{-5}
RMU	1.27	0.66	4.39	4	—	2.21×10^{-4}
ETW	1.31	0.033	0.16	—	2.58×10^{-5}	—
WGA	0.79	1.16	2.14	—	1.57×10^{-5}	1.57×10^{-5}
SatImp	0.49	0.87	1.43	0.17	1.98×10^{-5}	1.98×10^{-5}
ATWU	0.50	3.00	7.00	0.050	2.00×10^{-5}	1.50×10^{-5}

Table 6: Selected hyperparameters on TOFU. `1B` and `8B` are the learning rates used on `Llama-3.2-1B-Instruct` and `Llama-3.1-8B-Instruct`, respectively. The method-specific columns are overloaded across methods: β for DPO/NPO/SimNPO/WGA, β_1 for SatImp, τ (temperature) for ETW, c (steering coefficient) for RMU; δ for SimNPO, β_2 for SatImp, slr for ATWU, ℓ (target layer) for RMU. ETW was evaluated only on `1B` and RMU only on `8B`; the non-evaluated column is marked —.

Method	Shared			Method-Specific	
	lr	α	γ	β / β_1	δ / β_2 / slr
GradDiff	1.18×10^{-6}	0.80	0.12	—	—
DPO	1.18×10^{-5}	2.96	0.13	0.35	—
NPO	2.60×10^{-5}	0.63	4.76	0.07	—
SimNPO	2.08×10^{-5}	0.46	2.43	2.59	1.50
JensUn	1.54×10^{-6}	1.52	0.39	—	—
WGA	1.54×10^{-6}	0.22	2.79	0.40	—
SatImp	4.48×10^{-6}	0.23	2.60	2.64	0.20
ATWU	1.83×10^{-6}	0.54	2.68	2.25	0.020

Table 7: Selected hyperparameters on the canonical ten-subject RWKU batch. The learning rate is the one used on `Phi-3-Mini-4k-Instruct`.

repository defaults: AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$ and weight decay 0.01, a `cosine` learning-rate schedule with 20 warmup steps, gradient clipping at 1.0, and `fp16` precision. On WMDP, each run uses a single `NVIDIA A100-SXM4-80GB` GPU and trains for 150 optimizer steps at batch size 8 with 4 gradient-accumulation steps (effective batch size 32), using the `paged_adamw_8bit` optimizer without weight decay, a linear learning-rate decay without warmup, `bfloat16` precision, and gradient checkpointing. Each optimizer step uses a single forget domain, alternating between the two and starting with Cyber (Section C.2). We train every selected WMDP configuration with seeds 1–3.

For the final reported numbers, we evaluate each unlearned checkpoint on the full metric panel of Section B. In the main paper, we assess unlearning performance and retain-set preservation primarily via the baseline-relative summary metrics FQ, RD, and UQ introduced in Section B.6. These are our preferred headline metrics because they normalize forget-side gains and retain-side damage relative to the original checkpoint, making runs more directly comparable across models and forget sets. In particular, UQ serves as our primary one-number summary, since it rewards forgetting only to the extent that it is *not* purchased by collateral degradation on the retain set. On WMDP, where these summary metrics are not defined (Section B.5), we instead report WMDP-Bio, WMDP-Cyber, and WMDP accuracy together with MMLU, Rep., and WR, each as the mean $\pm$ sample standard deviation over three seeds.

Method	Shared			Method-Specific	
	lr	α	γ	β / β_1 / c	δ / β_2 / slr / ℓ
GradDiff	4.42×10^{-6}	1.14	1.06	—	—
NPO	1.33×10^{-5}	0.20	0.41	0.14	—
SimNPO	1.69×10^{-5}	0.27	0.44	2.28	2.00
JensUn	9.01×10^{-6}	0.67	2.50	—	—
RMU	3.45×10^{-5}	0.39	0.51	1.36	8
WGA	6.99×10^{-6}	0.19	0.80	0.19	—
SatImp	4.98×10^{-6}	0.34	4.68	1.06	0.10
ATWU	4.76×10^{-6}	0.10	10.0	2.39	0.0082

Table 8: Selected hyperparameters on WMDP with `Zephyr-7B-beta`, chosen on the preliminary corpus (Section C.3) and frozen for the final runs. For RMU, c is the steering coefficient and ℓ the unlearning layer.

To monitor broader post-unlearning behavior, we additionally report the utility metrics of Section B.3: MMLU, repetitiveness, and WR. We interpret these as *utility-preservation* probes rather than optimization targets. The goal of unlearning is not to improve general utility, but to preserve as much of the original model's utility as possible while removing the targeted knowledge. Accordingly, the relevant failure mode is utility *degradation*: drops in MMLU, drops in Rep. indicating more degenerate generation, and WR < 45 against the reference model. Occasional improvements relative to the reference model are not harmful and are therefore not penalized; we interpret them as benign side effects or measurement noise rather than as objectives of the unlearning procedure.

For completeness, the appendix also reports the underlying forget/retain metrics from Sections B.1 and B.2, namely the extraction-strength scores $\mathrm{ES}_{\mathcal{F}}$, $\mathrm{ES}_{\mathcal{R}}$, ES_{Δ} and the judge-based scores $\mathcal{J}_{\mathrm{P}}$, $\mathcal{J}_{\mathrm{ICR}}$, $\mathcal{J}_{\mathrm{W}}$, and $\mathcal{J}_{\mathrm{AVG}}$. Judge-based evaluation uses the fixed prompts provided in the supplementary material and is performed with OpenAI's `GPT-5.4 mini` judge model, instantiated as `gpt-5.4-mini` [OpenAI, 2026]. We validate this judge against human annotations in Fig. 7. MMLU is computed with the `lm-evaluation-harness` [Gao et al., 2024]. Main-paper tables report FQ, RD, UQ, MMLU, Rep., and WR, while the full metric panel is deferred to this section.

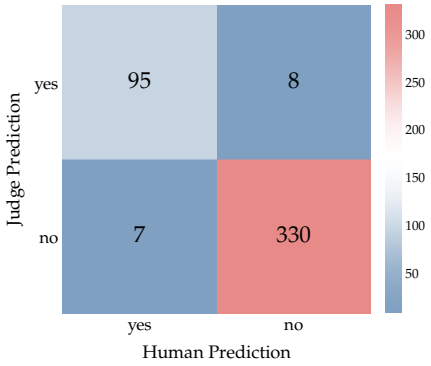


Figure 7: Agreement between the `GPT-5.4 mini` judge and human annotators on a 440-row sample of TOFU `forget01`. The judge matches the human label on 96.6% of calls; errors split as 8 false positives and 7 false negatives.

Judge reproducibility. A practical caveat is that OpenAI's hosted inference is not bit-exact reproducible: calling `gpt-5.4-mini` on the same input does not in general return the same completion or the same per-token log-probabilities, even at `temperature=0` and with a fixed `seed`.[14] This is a known property of the served models rather than of the API surface, and the standard mitigation in the community is to cache judged responses against a hash of the input rather than to expect run-to-run determinism. Because rejudging every checkpoint on every paraphrase carries non-trivial API cost (cf. the $180 figure quoted earlier in this section), we run the judge once per (checkpoint, query, paraphrase) tuple and report the resulting scores; the 96.6% judge–human agreement reported in Fig. 7 is computed on the same single-pass scores. In spot rejudging we observed only minor drift in $\mathcal{J}$-based scores across calls—small relative to the inter-method gaps in our headline tables—which we take as evidence that the rankings we report are robust to judge non-determinism.

[14] See e.g. the OpenAI developer-community threads at:
https://community.openai.com/t/reproducible-outputs-in-assistants-api/1167782 and
https://community.openai.com/t/logprobs-inconsistent-between-runs-for-4o/935082/2

Method	Surrogate			Paraphrase		Relative			Utility		
	$\mathrm{ES}_{\mathcal{F}}\downarrow$	$\mathrm{ES}_{\mathcal{R}}\uparrow$	$\mathrm{ES}_{\Delta}\uparrow$	$\mathcal{J}_{\mathrm{W}}\downarrow$	$\mathcal{J}_{\mathrm{AVG}}\uparrow$	FQ ↑	RD ↓	UQ ↑	MMLU	Rep.	WR
forget01 - Llama-3.2-1B-Instruct											
ORIGINAL	0.743	0.737	−0.007	90.0	41.5	0.0	0.0	0.0	45.1	559	50.0
GRADDIFF	0.220	0.625	0.405	77.5	35.9	13.9	13.6	0.3	45.0	563	51.5
DPO	0.134	0.566	0.432	**17.5**	38.1	**80.6**	8.3	**72.2**	45.1	552	56.5
NPO	0.085	0.539	0.454	45.0	37.7	50.0	9.3	40.7	45.3	562	59.0
SIMNPO	0.087	0.574	0.487	52.5	**38.3**	41.7	7.9	33.7	45.4	565	54.0
JENSUN	0.295	0.600	0.304	60.0	29.2	33.3	29.8	3.5	44.9	555	55.0
WGA	0.093	**0.675**	**0.582**	80.0	34.6	11.1	16.7	0.0	45.2	560	58.5
SATIMP	0.084	0.619	0.536	55.0	**38.3**	38.9	**7.8**	31.1	45.1	560	56.0
ATWU	**0.046**	0.604	0.558	52.5	36.5	41.7	12.1	29.5	45.1	557	52.0
forget05 - Llama-3.2-1B-Instruct											
ORIGINAL	0.727	0.737	0.009	90.5	42.9	0.0	0.0	0.0	45.1	559	50.0
GRADDIFF	0.178	0.620	0.442	68.6	36.1	24.2	15.8	8.4	44.8	563	50.5
DPO	0.205	0.624	0.419	50.5	38.6	44.2	10.0	34.2	44.8	551	45.0
NPO	0.100	0.644	0.543	47.5	38.4	47.5	10.6	36.9	45.0	565	53.5
SIMNPO	0.121	0.689	0.568	54.6	38.8	39.7	9.6	30.1	45.1	568	54.5
JENSUN	0.047	0.208	0.161	**0.0**	1.1	**100.0**	97.3	2.7	45.4	369	25.0
WGA	0.076	**0.746**	**0.670**	53.0	**40.2**	41.5	**6.4**	35.1	45.0	568	49.0
SATIMP	0.061	0.690	0.629	47.9	38.3	47.1	10.9	36.3	44.9	569	57.0
ATWU	**0.041**	0.696	0.655	37.5	37.1	58.6	13.6	**45.0**	45.1	574	60.5
forget10 - Llama-3.2-1B-Instruct											
ORIGINAL	0.706	0.737	0.030	93.1	43.0	0.0	0.0	0.0	45.1	559	50.0
GRADDIFF	0.107	0.658	0.550	57.9	37.4	37.8	13.0	24.8	45.2	563	55.5
DPO	0.194	0.716	0.523	53.3	38.2	42.8	11.2	31.6	44.9	560	59.0
NPO	0.103	0.730	0.627	41.0	**42.4**	56.0	**1.4**	54.6	44.9	563	55.0
SIMNPO	0.106	0.738	0.632	57.4	41.1	38.3	4.4	33.9	45.1	566	57.0
JENSUN	**0.033**	0.570	0.537	**1.8**	36.4	**98.0**	15.3	**82.8**	45.1	343	23.5
ETW	0.108	0.679	0.572	42.8	39.0	54.1	9.3	44.8	45.5	561	54.5
WGA	0.062	**0.759**	0.696	46.6	39.3	50.0	8.5	41.5	45.0	568	56.0
SATIMP	0.043	0.747	0.704	26.9	41.2	71.2	4.1	67.1	44.8	570	55.0
ATWU	0.035	0.753	**0.717**	14.5	40.3	84.4	6.3	78.1	45.0	575	51.5

Table 9: TOFU full-panel results for `Llama-3.2-1B-Instruct` across the three forget splits. In each performance column, best is in bold and second-best underlined. ETW was evaluated only on `forget10`.

The importance of proper tuning. Reported baseline performance in the LLM-unlearning literature is unusually sensitive to per-method hyperparameter choices, and numbers across papers are not always directly comparable. RMU [Li et al., 2024] is the clearest example: Wang et al. [2025b] report it underperforming several preference-based baselines on TOFU, yet under our uniform two-stage tuning protocol it reaches UQ $= 85.9$ on `forget10` with `Llama-3.1-8B-Instruct`, behind only ATWU (UQ $= 91.7$) and JENSUN (UQ $= 88.3$), and well ahead of every preference-based method (Table 1). The driver is RMU's idiosyncratic configuration space, where a mis-specified unlearning layer ℓ alone can significantly swing UQ; our Bayesian search lands on $\ell = 4$, $c = 4.39$ (Table 6), which differs substantially from the configurations typically reported for WMDP-style settings. The relevant tuning gap is not the loss coefficients α, γ alone: Wang et al. [2025b] and Yang et al. [2025] do tune those in narrow paper-seeded ranges, but the learning rate (which is not equivalent to a coefficient rescaling under `AdamW` with gradient clipping and a non-constant LR schedule) and, in RMU's case, the steering coefficient and target layer, which are not loss coefficients at all. By

Method	Surrogate			Paraphrase		Relative			Utility		
	$ES_{\mathcal{F}}\downarrow$	$ES_{\mathcal{R}}\uparrow$	$ES_{\Delta}\uparrow$	$\mathcal{J}_{W}\downarrow$	$\mathcal{J}_{AVG}\uparrow$	FQ ↑	RD ↓	UQ ↑	MMLU	Rep.	WR
forget01 - Llama-3.1-8B-Instruct											
ORIGINAL	0.977	0.992	0.015	100.0	68.2	0.0	0.0	0.0	66.6	546	50.0
GRADDIFF	0.203	0.724	0.521	72.5	50.9	27.5	25.3	2.2	66.6	529	45.0
DPO	0.219	0.805	0.585	**12.5**	56.6	**87.5**	16.9	**70.6**	66.3	482	36.0
NPO	0.093	0.700	0.607	32.5	53.3	67.5	21.8	45.7	66.3	544	48.5
SIMNPO	0.080	0.666	0.586	27.5	51.4	72.5	24.6	47.9	66.4	541	50.0
JENSUN	0.887	**0.979**	0.092	95.0	57.6	5.0	15.6	0.0	66.5	278	19.5
WGA	0.080	0.907	**0.827**	42.5	63.0	57.5	7.5	50.0	66.5	534	48.5
SATIMP	0.083	0.809	0.726	42.5	56.5	57.5	17.1	40.4	66.6	525	41.5
ATWU	**0.065**	0.885	0.820	45.0	**64.8**	55.0	**5.0**	50.0	66.7	558	57.0
forget05 - Llama-3.1-8B-Instruct											
ORIGINAL	0.972	0.992	0.020	99.0	68.6	0.0	0.0	0.0	66.6	546	50.0
GRADDIFF	0.256	0.782	0.525	75.0	51.6	24.2	24.7	0.0	66.7	544	45.5
DPO	0.268	0.815	0.548	41.5	58.4	58.1	14.8	43.3	65.1	530	34.0
NPO	0.123	0.875	0.752	50.0	60.6	49.5	11.7	37.8	66.3	549	53.0
SIMNPO	0.102	0.852	0.750	42.5	55.7	57.1	18.7	38.4	66.0	545	56.5
JENSUN	**0.033**	0.924	**0.891**	**1.0**	61.2	**99.0**	10.7	**88.3**	65.2	474	29.0
WGA	0.058	**0.931**	0.874	24.5	**63.6**	75.3	**7.2**	68.0	66.6	541	45.5
SATIMP	0.035	0.891	0.855	15.5	61.0	84.3	11.0	73.4	66.6	546	56.0
ATWU	**0.033**	0.914	0.881	11.7	61.7	88.2	10.0	78.2	66.8	541	48.5
forget10 - Llama-3.1-8B-Instruct											
ORIGINAL	0.979	0.992	0.013	99.5	67.8	0.0	0.0	0.0	66.6	546	50.0
GRADDIFF	0.096	0.838	0.742	60.0	56.1	39.7	17.3	22.4	66.1	545	48.0
DPO	0.314	0.860	0.547	40.2	57.6	59.5	15.0	44.5	64.3	533	38.5
NPO	0.129	0.927	0.798	51.7	64.4	48.0	5.0	43.0	66.2	551	48.5
SIMNPO	0.077	0.905	0.828	30.5	63.4	69.3	6.5	62.8	65.8	541	50.5
JENSUN	**0.033**	0.942	0.910	**2.0**	61.3	**98.0**	9.6	88.3	65.3	236	11.0
WGA	0.046	0.970	0.924	33.0	64.6	66.8	4.8	62.1	66.0	550	60.0
SATIMP	0.035	0.952	0.917	21.2	65.8	78.6	3.0	75.6	65.9	549	53.5
RMU	**0.033**	**0.989**	**0.956**	12.8	**66.9**	87.2	**1.3**	85.9	65.3	540	50.5
ATWU	**0.033**	0.973	0.940	4.8	65.4	95.2	3.5	**91.7**	66.5	558	58.0

Table 10: TOFU full-panel results for `Llama-3.1-8B-Instruct` across the three forget splits. In each performance column, best is in bold and second-best underlined. RMU was evaluated only on `forget10`.

contrast, Singh et al. [2025] and Dorna et al. [2025] grid-search the learning rate per method and report RMU as competitive, consistent with our finding. We view this not as a critique of any specific prior comparison but as empirical evidence that fair ranking across this literature requires re-tuning each method for the target setting, particularly when methods exposing many interacting knobs (RMU, SIMNPO, SATIMP) are compared against methods with very few (GRADDIFF, JENSUN). We therefore tune every baseline uniformly via Bayesian optimization, seeded with each paper's recommended defaults; the full protocol and selected configurations are reported in Sections C.3 and C.4.

TOFU full-panel results. Tables 9 and 10 report the full TOFU metric panels across all forget splits and both base models. On `forget05` and `forget10`, ATWU attains the highest UQ at both model scales once utility-collapsing runs are excluded. This distinction is important: aggressive methods such as JENSUN can obtain very strong forget-side scores, but often do so by severely damaging

Method	Surrogate			Paraphrase		Relative			Utility		
	R-L$_{\mathcal{F}}$ ↓	R-L$_{\mathcal{N}}$ ↑	N_{Δ} ↑	$\mathcal{J}_{\mathrm{W}}$ ↓	$\mathcal{J}_{\mathrm{AVG}}$ ↑	FQ ↑	RD ↓	UQ ↑	MMLU	Rep.	WR
ORIGINAL	0.683	0.759	0.077	84.8	65.8	0.0	0.0	0.0	70.1	529	50.0
GRADDIFF	0.053	0.445	0.392	15.3	41.6	81.9	36.7	45.2	69.2	514	43.0
DPO	0.499	**0.590**	0.090	74.4	**54.6**	12.3	**17.0**	0.0	69.7	523	43.0
NPO	0.236	0.541	0.305	44.4	40.1	47.6	39.1	8.5	68.1	543	33.5
SIMNPO	0.218	0.561	0.343	40.9	46.2	51.8	29.7	22.1	68.8	529	45.5
JENSUN	0.062	0.434	0.372	**12.4**	42.0	**85.4**	36.1	49.3	70.2	523	46.5
WGA	0.061	0.515	0.454	18.7	48.6	78.0	26.1	51.9	69.4	518	41.0
SATIMP	0.069	0.533	0.464	19.3	46.5	77.3	29.3	48.0	69.8	518	40.0
ATWU	**0.050**	0.562	**0.512**	15.8	50.9	81.4	22.7	**58.7**	70.3	519	46.0

Table 11: RWKU canonical ten-subject batch, `Phi-3-Mini-4k-Instruct`. In each performance column, best is in bold and second-best underlined.

generation quality, as reflected by large drops in Rep. and WR. We therefore interpret the tables through the joint lens of forgetting, retain preservation, and general utility.

The main limitation appears on the smaller forget splits, especially `forget01`. In these settings, the scorer receives relatively few forget examples and fewer effective scorer updates, making it harder to learn a stable forget-specificity signal. `forget01` contains only 40 question–answer pairs, which is below the forget-set size at which a scorer trained in isolation learns a useful ranking (Section D.3); `forget05`, with 200 pairs, is already above it. As the forget set becomes larger, the learned scorer has more opportunity to separate forget-specific tokens from structural context, and the advantage of ATWU becomes clearer. This trend is most visible on `forget10`, where ATWU achieves the strongest or near-strongest UQ while preserving MMLU, Rep., and WR close to the original checkpoint. These results suggest that the learned token-weighting mechanism is most beneficial when enough forget-side signal is available for the scorer to train reliably.

RWKU full-panel results. Table 11 reports the corresponding full metric panel on the canonical ten-subject RWKU batch. This setting is challenging for a different reason than TOFU. In RWKU, the forget set consists of paragraphs about the target entities, whereas evaluation is performed through separate queries about those entities. Thus, the unlearning objective is not optimized on the same question–answer format used at evaluation time; the model must forget information expressed in paragraph form and generalize that forgetting to query-based probes. Despite this mismatch, ATWU achieves the highest UQ, improving over the strongest non-ATWU baseline while preserving MMLU and maintaining a non-collapsed win rate against the original model. The ROUGE-L surrogate metrics show a similar trend, with ATWU attaining the best N_{Δ}, but the judge-based metrics provide the more

Method	QA ↓	FB ↓	AA ↓
ORIGINAL	77.5	87.7	89.2
GRADDIFF	8.5	14.6	23.0
DPO	66.9	78.6	77.7
NPO	38.7	49.7	44.9
SIMNPO	36.6	40.4	45.7
JENSUN	**6.3**	**8.2**	22.6
WGA	12.9	21.1	22.0
SATIMP	11.3	19.9	26.6
ATWU	10.1	17.5	**19.8**

Table 13: $\mathcal{J}_{\mathrm{W}}$ per RWKU probe type. Best in bold, second-best underlined.

Method	Surrogate			Paraphrase		Relative				Utility		
	R-L$_{\mathcal{F}}$ ↓	R-L$_{\mathcal{N}}$ ↑	N_{Δ} ↑	$\mathcal{J}_{\mathrm{W}}$ ↓	$\mathcal{J}_{\mathrm{AVG}}$ ↑	FQ ↑	RD ↓	UQ ↑	ΔUQ ↑	MMLU	Rep.	WR
ORIGINAL	0.683	0.759	0.077	84.8	65.8	0.0	0.0	0.0	—	70.1	529	50.0
ATWU$_{\mathrm{DPO}}$	0.501	**0.657**	0.156	68.2	**57.1**	19.6	**13.2**	6.4	+6.4	69.5	512	36.0
ATWU$_{\mathrm{NPO}}$	0.205	0.490	0.285	39.1	42.8	53.8	34.9	18.9	+10.4	67.7	545	35.5
ATWU$_{\mathrm{SimNPO}}$	0.109	0.567	0.458	32.6	50.1	61.5	23.8	37.8	+15.7	69.9	531	43.5
ATWU	**0.050**	0.562	**0.512**	**15.8**	50.9	**81.4**	22.7	**58.7**	+6.8	70.3	519	46.0

Table 12: RWKU canonical ten-subject batch with ATWU instantiated using different forget losses, `Phi-3-Mini-4k-Instruct`. ΔUQ reports the gain over the corresponding unweighted objective in Table 11. In each performance column best is in bold and second-best underlined.

Method	Forget			Utility		
	Bio ↓	Cyber ↓	WMDP ↓	MMLU	Rep.	WR
ORIGINAL	64.7	44.3	54.5	58.5	625	50.0
GRADDIFF	29.5 ± 3.5	25.8 ± 1.3	27.6 ± 2.3	55.2 ± 1.1	457 ± 167	34.2 ± 22.9
NPO	61.5 ± 0.4	39.1 ± 1.1	50.3 ± 0.7	56.9 ± 0.2	636 ± 1	57.7 ± 2.8
SIMNPO	45.0 ± 4.7	32.4 ± 1.5	38.7 ± 3.0	55.7 ± 0.3	633 ± 8	40.8 ± 6.0
JENSUN	42.1 ± 8.3	36.2 ± 3.5	39.1 ± 4.7	58.0 ± 0.2	475 ± 115	43.0 ± 17.0
RMU	30.3 ± 2.4	25.5 ± 0.5	27.9 ± 1.4	56.2 ± 0.1	630 ± 1	54.3 ± 2.8
WGA	26.5 ± 1.6	26.0 ± 1.3	26.3 ± 1.0	56.7 ± 0.5	575 ± 4	54.8 ± 3.8
SATIMP	39.8 ± 2.7	36.3 ± 0.4	38.0 ± 1.4	58.3 ± 0.2	625 ± 3	60.2 ± 3.3
ATWU	27.9 ± 0.9	27.6 ± 1.8	27.8 ± 0.5	57.6 ± 0.4	590 ± 17	49.0 ± 1.7

Table 14: WMDP results with `Zephyr-7B-beta`, reported as mean $\pm$ sample standard deviation over seeds 1–3 on the reproducible corpus of Appendix C.2. Bio and Cyber are multiple-choice accuracies on WMDP-Bio and WMDP-Cyber, and WMDP is their mean; chance is 25% (Section B.5). Materially degraded utility values (MMLU drop $>$ 2pp, Rep. drop $>$ 6%, or WR $<$ 45) are shown in red.

important comparison: methods with strong forget-side scores can still suffer substantial retain-side or utility degradation. Overall, the RWKU results indicate that learned token-level reweighting remains effective even when the forget data and evaluation queries differ in format.

RWKU results per probe type. The RWKU forget scores above pool the FB, QA, and AA probes (Section B.4). Table 13 breaks $\mathcal{J}_{\mathrm{W}}$ down by probe type. ATWU attains the lowest score on the adversarial-attack probes (19.8). JENSUN attains the lowest scores on the direct QA and FB probes (6.3 and 8.2), but its advantage does not carry over to adversarial prompting (22.6 on AA).

ATWU with different forget losses. Finally, Table 12 evaluates whether ATWU is tied to a particular forget loss. In the main ATWU variant, we use the SATGA$^+$ forget loss. Here, we compare with alternative forget loss choices derived from DPO, NPO, and SIMNPO, integrating the learned scorer into the forget-side term of each objective as described in Section D.4. This keeps the central mechanism of ATWU fixed – learning token-level forget-specificity scores and using them to weight the forget update – while only changing the underlying forget loss. Across all forget losses, ATWU improves UQ relative to the corresponding unweighted objective. The strongest results are obtained with the SATGA$^+$ instantiation, but the gains for DPO, NPO, and SIMNPO show that the learned scorer is not specific to one loss form. Rather, ATWU provides a general way to turn sequence-level forget objectives into selective token-weighted objectives.

WMDP results. Table 14 reports the WMDP results. WGA, GRADDIFF, ATWU, and RMU bring WMDP accuracy within three points of the 25% chance level (26.3, 27.6, 27.8, and 27.9). Among them, ATWU is the only method that triggers no utility flag: it loses less than one point of MMLU and 5.6% of Rep., whereas WGA loses 8% of Rep., RMU more than two points of MMLU, and GRADDIFF degrades on all three utility measures. SATIMP and NPO preserve utility but forget less (38.0 and 50.3). During hyperparameter selection on the preliminary corpus, ATWU and WGA were also the only methods that met both selection thresholds on every seed (Section C.3).

C.6 Computational Cost

ATWU updates the scorer once every five model updates (Section D.5). The scorer update itself is inexpensive, but computing and applying the token-level weights adds wall-clock cost. Table 15 reports the average time to process one forget/retain batch and the peak GPU memory of GRADDIFF, SATIMP, and ATWU; for ATWU, the averages include the scheduled scorer updates. All measurements are taken on a single `NVIDIA A100` GPU with 80 GB of memory, using batch size 4 and sequence length 512 for both the forget and the retain batches. This is a controlled benchmark setting rather than the training configuration of our final runs.

Model	GradDiff	SatImp	ATWU
`Llama-3.2-1B-Instruct`	248 ms / 16.2 GB	287 ms / 17.5 GB	758 ms / 19.4 GB
`Llama-3.1-8B-Instruct`	1078 ms / 59.5 GB	1085 ms / 73.4 GB	1599 ms / 63.4 GB
`Zephyr-7B-beta`	1027 ms / 54.4 GB	1027 ms / 67.6 GB	1155 ms / 56.1 GB

Table 15: Average time per forget/retain batch and peak GPU memory (time / memory) on a single A100 GPU with batch size 4 and sequence length 512.

Judge / rubric	Forget ↓		Retain ↑		ATWU advantage
	WGA	ATWU	WGA	ATWU	in Δ ↑
`Qwen2.5-14B`, target	6.9	6.6	61.4	62.7	+1.6
`Claude Opus 5`, target	12.1	11.6	64.9	67.1	+2.7
`Claude Opus 5`, ground truth	6.6	6.3	48.3	50.2	+2.2

Table 17: WGA and ATWU on RWKU under three judge and rubric combinations. "Target" is the alternative rubric of Table 16; "ground truth" is its variant with the ground-truth answer.

Relative to GradDiff, ATWU takes 3.1×, 1.48×, and 1.12× as long per batch on `Llama-3.2-1B-Instruct`, `Llama-3.1-8B-Instruct`, and `Zephyr-7B-beta`, respectively. On the two Llama models, the absolute overhead is similar, at approximately 0.51 s per batch, so the relative overhead decreases as the model grows; the 3.1× ratio on the 1B model mainly reflects its short base step time. Peak memory exceeds that of GradDiff by 1.7–3.9 GB and stays below that of SatImp on the two larger models. The scorer itself is therefore lightweight; token-level weighting adds a roughly constant absolute cost, which is proportionally largest on the smallest model.

C.7 Sensitivity to the Judge Model and Rubric

Our judge-based metrics rely on a single judge, `GPT-5.4 mini`, and a single rubric (Section B.2). To test whether our conclusions depend on this choice, we re-score the saved RWKU generations with an alternative judge script that uses `Qwen2.5-14B-Instruct` [Qwen Team et al., 2024] as the judge. Our rubric gives the judge the ground-truth answer and asks for a binary decision on whether the specific fact has been recovered (on RWKU, ignoring the judge's own knowledge of the entity). The alternative script instead gives the judge only the question, the model's answer, and the unlearning target, and asks for a score in $[0, 1]$ reflecting how much the answer still reflects that target.

Method	Forget ↓	Retain ↑	Δ ↑
Original	84.5	85.5	1.0
GradDiff	9.0	55.0	46.0
DPO	67.7	**74.5**	6.8
NPO	21.8	59.9	38.1
SimNPO	24.7	<u>65.3</u>	40.7
JensUn	**6.0**	48.3	42.3
WGA	6.9	61.4	<u>54.5</u>
SatImp	16.4	61.7	45.4
ATWU	<u>6.6</u>	62.7	**56.1**

Table 16: `Qwen2.5-14B` judge scores on RWKU. Best in bold, second-best underlined.

Table 16 reports the results. ATWU again obtains the best forget–retain trade-off, measured by Δ = Retain − Forget. JensUn obtains the lowest forget score, but with substantially lower retain performance. Among methods that maintain strong retention, WGA is the closest competitor, with a slightly higher forget score (6.9 vs. 6.6) and a lower retain score (61.4 vs. 62.7).

Because the alternative rubric does not see the ground-truth answer, it can reward on-topic responses that do not recover the specific fact. As an additional check, we therefore re-evaluate the two strongest methods, WGA and ATWU, with `Claude Opus 5` (queried through the Anthropic API) as the judge, using both the alternative rubric and a ground-truth variant of it that additionally provides the ground-truth answer and asks whether the corresponding fact is recovered (Table 17). The margin between the two methods is small, but both the forget and the retain components favor ATWU under all three evaluations.

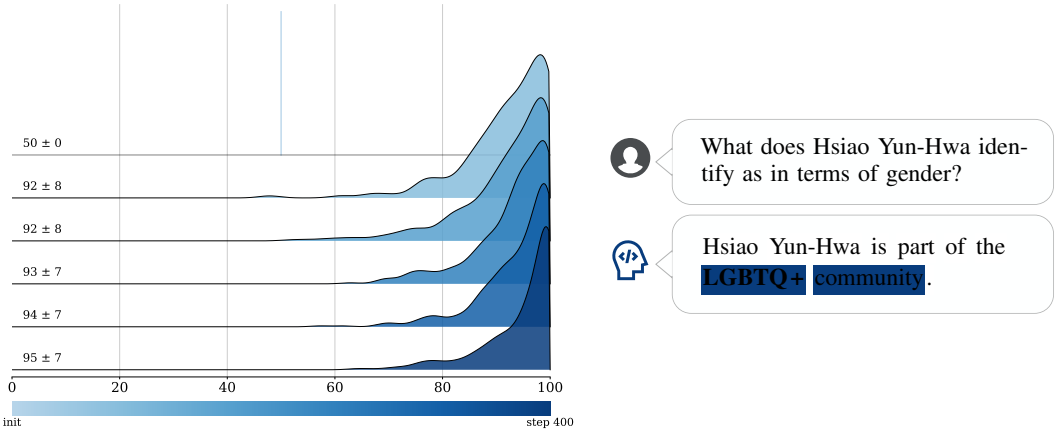
D Additional Experiments

This section provides additional experimental details and ablations that support the main findings. We first ask whether a single linear function of the model's hidden states can even, in principle, separate forget-specific tokens from structural ones, with and without supervision. We then test whether the learned scorer generalizes to forget examples it was not trained on, and how much forget data it needs to learn such a separation. Next, we show how ATWU can be combined with alternative forget losses by inserting the learned scorer only into the forget-side token-level terms. Finally, we ablate the main scorer design choices, including its regularization, training state, and update frequency, and measure the sensitivity of ATWU to its budget ρ and saturation parameter β. Together, these experiments test whether the gains of ATWU come from a particular objective choice, or from the more general mechanism of learning where the forget update should be applied.

D.1 Linear Separability of Informative Tokens

Before validating our parametric scorer in detail, we briefly ask whether a linear function of the model's hidden states can even, in principle, separate forget-specific tokens from structural ones, and whether such a scorer can be learned without supervision. The two experiments below answer affirmatively in both regimes.

Supervised baseline. We train a single-layer linear scorer on the final-layer hidden states of `Llama-3.1-8B-Instruct` fine-tuned on TOFU using binary cross-entropy against the ground-truth token-level forget labels of Zhou et al. [2026], for one epoch on `forget10`. Figure 8 reports the per-sample AUROC distribution as training progresses, together with the final scorer's output on a representative forget sample. After only $\sim 20\%$ of one epoch the AUROC distribution is already concentrated at the high end, and the final scorer cleanly recovers the GT forget tokens on the qualitative example. A single linear direction on the model's hidden states therefore suffices to separate forget-specific tokens, at least *when the labels are available*.



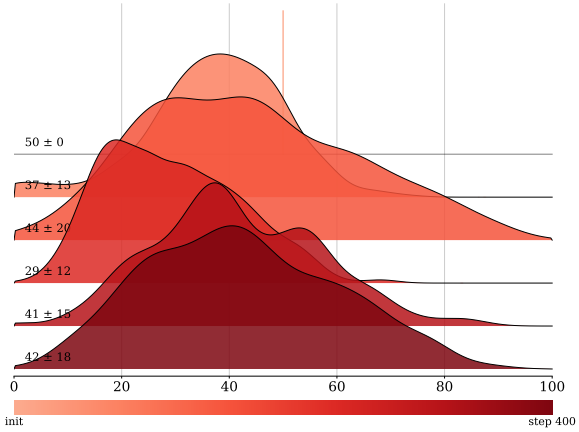
(a) Per-sample AUROC distribution of the BCE-trained scorer over training.

(b) Final scorer output on representative prompt.

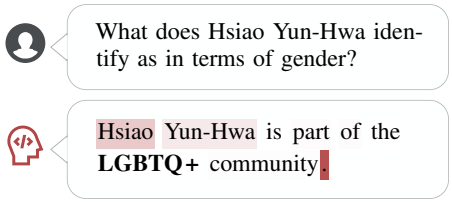
Figure 8: **Supervised baseline.** A linear scorer trained with binary cross-entropy against the GT labels of Zhou et al. [2026] cleanly recovers the forget span after a fraction of an epoch. Bold tokens mark the ground-truth forget-specific span.

Unsupervised scorer with the unlearning objective. Ground-truth token labels of the kind released by Zhou et al. [2026] require manual or LLM-based annotation of the forget data, which is unrealistic for many unlearning requests, especially when the forget set contains private, copyrighted, or otherwise sensitive content. We therefore ask whether the retain-conflict objective used inside ATWU (Eq. 7) can recover a comparable linear scorer without labels. Figure 9 reports the analogous trajectories and qualitative output in two configurations: training the scorer against the original TOFU-fine-tuned target model that has *memorized* the forget content (top row), and against a *retain*

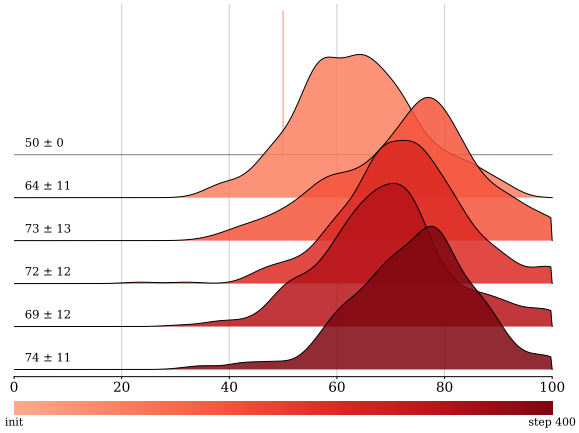
model with the same architecture fine-tuned only on the retain split, with no exposure to the forget set (bottom row). On the target model, the unsupervised objective fails to localize the forget span; its near-perfect predictions on memorized forget content provide too little gradient signal. On the retain model, by contrast, the same objective recovers the answer-bearing tokens (*community*, *Hsiao*, *Yun-Hwa*, partially *LGBTQ*), comparable in shape to the supervised scorer. This observation underpins a central design choice in ATWU: jointly training the scorer with the unlearned model gradually exposes the scorer to representations closer to the retain regime, supplying the gradient signal needed to localize forget-specific tokens without any labels.



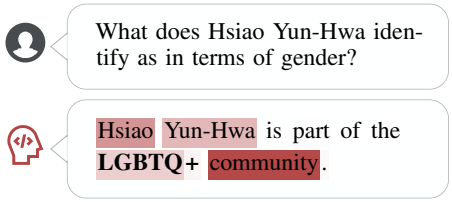
(a) ATWU objective on the *target* (memorized) model.



(b) Final output (target model).



(c) ATWU objective on the *retain* model.



(d) Final output (retain model).

Figure 9: **Unsupervised scorer.** Trained against the original target model (top), the ATWU objective fails to localize the GT forget span; trained against a retain model (bottom), the same objective recovers the answer-bearing tokens. Bold tokens mark the ground-truth forget-specific span.

D.2 Scorer Generalization to Held-Out Forget Data

The scorer is a single linear function shared across all forget examples. We test whether it generalizes beyond the examples it was trained on. We run ATWU on TOFU `forget10` with `Llama-3.2-1B-Instruct`, seed 0, and the selected configuration of Table 6, but hold out one fifth of the 400 forget examples from training. We consider two hold-out schemes with five folds each. *Unseen authors* holds out four complete authors (all 20 questions about each) and tests generalization to new entities. *Unseen facts* holds out four questions about every author and tests generalization to new facts about entities seen during training. After unlearning, we apply each learned scorer to the hidden states of the original TOFU model on all 400 forget examples and compute the per-example token-ranking AUROC against the ground-truth labels of Zhou et al. [2026], skipping the 12 exam-

ples whose labels are all positive or all negative. We then average the AUROC separately over the examples seen during training (about 310 per fold) and the held-out examples (about 78 per fold).

Fold	Unseen authors			Unseen facts		
	Seen	Unseen	Gap	Seen	Unseen	Gap
1	0.746	0.760	−0.014	0.766	0.769	−0.002
2	0.728	0.719	+0.009	0.752	0.720	+0.032
3	0.725	0.741	−0.015	0.744	0.741	+0.003
4	0.739	0.721	+0.018	0.728	0.717	+0.011
5	0.772	0.747	+0.025	0.736	0.694	+0.042
Mean	0.742 ± 0.019	0.738 ± 0.017	$+0.004 \pm 0.018$	0.745 ± 0.015	0.728 ± 0.028	$+0.017 \pm 0.019$

Table 18: Token-ranking AUROC of the ATWU scorer on forget examples seen during training and on held-out examples, TOFU `forget10`, `Llama-3.2-1B-Instruct`. Gap is seen minus unseen AUROC; the last row reports mean ± standard deviation over the five folds.

Table 18 reports the results. Under both schemes, AUROC on held-out examples is close to AUROC on seen examples: the mean gap is $+0.004$ for unseen authors and $+0.017$ for unseen facts, in both cases smaller than the fold-to-fold standard deviation of the gap. The learned scorer therefore transfers both to new entities and to new facts about entities it has seen.

D.3 Forget-Set Size and Scorer Learning

The unsupervised experiment of Section D.1 trains the scorer on the full `forget10` split. To measure how much forget data the scorer needs, we isolate scorer learning from unlearning: we freeze the model parameters at the retain-trained TOFU checkpoint of `Llama-3.2-1B-Instruct` and train only the scorer, with the ATWU objective, on increasingly large subsets of the forget set. Figure 10 reports the resulting token-ranking AUROC against the ground-truth labels of Zhou et al. [2026].

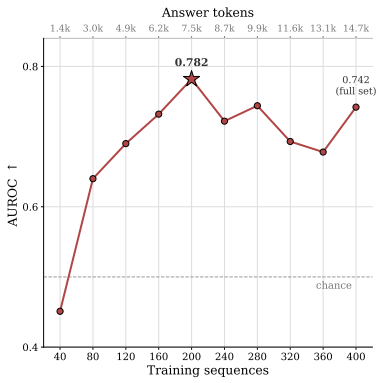


Figure 10: Token-ranking AUROC of the ATWU scorer, trained with the model frozen, versus the number of `forget10` training sequences (bottom) and answer tokens (top).

With 40 sequences (about 1.4k tokens), the scorer does not learn a useful ranking (AUROC 0.45). A meaningful signal appears at 80–120 sequences (about 3–5k tokens, AUROC 0.64–0.69) and becomes stronger at 160–200 sequences (0.73–0.78). Beyond this point, AUROC fluctuates between 0.68 and 0.74, suggesting that the diversity of the forget examples matters as well as their number. We therefore do not observe a universal threshold, but in this setting roughly 80–120 sequences suffice for useful scorer learning. This is consistent with the weaker results of ATWU on `forget01`, which contains 40 question–answer pairs (Section C.5). For smaller forget sets, the available signal could be increased by enriching the forget set with paraphrases or other meaning-preserving variants of its examples.

D.4 ATWU with Different Forget Losses

ATWU is not tied to a single forget loss. Whenever an unlearning objective contains a forget-side term that decomposes autoregressively over tokens, we can replace that term by its scorer-weighted analog while leaving the rest of the objective unchanged. This preserves the baseline loss structure but redirects the forget update toward to-

Method	Replacement	Scope
DPO	$r(\mathrm{x}^-) \mapsto r^g(\mathrm{x}^-)$	dispreferred sequence only
NPO	$r(\mathrm{x}) \mapsto r^g(\mathrm{x})$	forget sequence
SimNPO	$\hat{\ell}(\mathrm{x}) \mapsto \hat{\ell}^g(\mathrm{x})$	length-normalized forget NLL

Table 19: Summary of how the learned ATWU scorer is integrated into each forget loss.

kens that the scorer identifies as forget-specific.

We use this recipe to integrate the DPO, NPO, and SIMNPO losses into ATWU, evaluated in Fig. 1b and Table 3. Retain-side terms, preferred responses, and frozen reference-model terms are never reweighted; only the original forget sequence receives scorer-weighted token mass. Table 19 gives the corresponding replacements, and the following paragraphs define them precisely.

Notation and reweighted building blocks. For a sequence $\mathrm{x} \in \mathcal{V}^\star$ and model parameters $\boldsymbol{\theta}$, define the unnormalized autoregressive negative log-likelihood and its scorer-reweighted counterpart as

$$\ell(\boldsymbol{\theta}\,;\,\mathrm{x}) \;=\; -\sum_{t=1}^{|\mathrm{x}|} \log p_{\boldsymbol{\theta}}(\mathrm{x}_t \mid \mathrm{x}_{<t}), \qquad \ell^g(\boldsymbol{\theta}\,;\,\mathrm{x}) \;=\; -\sum_{t=1}^{|\mathrm{x}|} g_t \,\log p_{\boldsymbol{\theta}}(\mathrm{x}_t \mid \mathrm{x}_{<t}), \tag{22}$$

where $g_t = g_{\mathbf{w}}(\mathbf{h}_{\boldsymbol{\theta}}(\mathrm{x}_t))$ is the scorer output at position t. Dataset-level losses are obtained by summing over sequences:

$$\mathcal{L}(\boldsymbol{\theta}\,;\,\mathcal{D}) = \sum_{\mathrm{x}\in\mathcal{D}} \ell(\boldsymbol{\theta}\,;\,\mathrm{x}), \qquad \mathcal{L}^g(\boldsymbol{\theta}\,;\,\mathcal{D}) = \sum_{\mathrm{x}\in\mathcal{D}} \ell^g(\boldsymbol{\theta}\,;\,\mathrm{x}).$$

For methods that compare a trainable unlearned model $\boldsymbol{\theta}_{\mathrm{un}}$ against a frozen reference model $\boldsymbol{\theta}_{\mathrm{ref}}$, we define the sequence-level log-ratio and its scorer-reweighted analog as

$$r_{\boldsymbol{\theta}_{\mathrm{un}}}(\mathrm{x}) \;=\; -\ell(\boldsymbol{\theta}_{\mathrm{un}}\,;\,\mathrm{x}) + \ell(\boldsymbol{\theta}_{\mathrm{ref}}\,;\,\mathrm{x}), \qquad r^g_{\boldsymbol{\theta}_{\mathrm{un}}}(\mathrm{x}) \;=\; -w_{\mathrm{x}}\,\ell^g(\boldsymbol{\theta}_{\mathrm{un}}\,;\,\mathrm{x}) + \ell(\boldsymbol{\theta}_{\mathrm{ref}}\,;\,\mathrm{x}), \tag{23}$$

where

$$w_{\mathrm{x}} = \frac{|\mathrm{x}|}{\sum_{t=1}^{|\mathrm{x}|} g_t}.$$

This normalization keeps the total token mass of the trainable-model term equal to that of the original sequence loss, since $\sum_t w_{\mathrm{x}} g_t = |\mathrm{x}|$. The scorer therefore redistributes the contribution across token positions without changing the average scale of the trainable-model term. The reference term is left unchanged because the reference model is frozen and does not depend on the scorer.

Each augmented objective below replaces only the forget-side building block of the original loss by its scorer-reweighted counterpart, such as ℓ^g, r^g, or a scorer-modulated saturation term.

DPO [Rafailov et al., 2023]. DPO for unlearning operates on a paired forget set $\mathcal{A} = \{(\mathrm{x}_i^+, \mathrm{x}_i^-)\}_{i=1}^N$, where x_i^- is the original forget sequence and x_i^+ is an alternative sequence that the model should prefer. The original loss compares the two sequences through a sigmoid of their log-ratio difference:

$$\mathcal{L}_{\mathrm{DPO}}(\boldsymbol{\theta}_{\mathrm{un}}\,;\,\mathcal{A}) \;=\; -\frac{2}{\beta} \sum_{(\mathrm{x}^+,\mathrm{x}^-)\in\mathcal{A}} \log\sigma\Big(\beta\big[r_{\boldsymbol{\theta}_{\mathrm{un}}}(\mathrm{x}^+) - r_{\boldsymbol{\theta}_{\mathrm{un}}}(\mathrm{x}^-)\big]\Big). \tag{24}$$

The two sides of each pair play asymmetric roles. The dispreferred sequence x^- contains the information to be forgotten, whereas the preferred sequence x^+ is intended to remain ordinary, acceptable language. We therefore apply the scorer only to the dispreferred side:

$$\mathcal{L}^g_{\mathrm{DPO}}(\boldsymbol{\theta}_{\mathrm{un}}\,;\,\mathcal{A}) \;=\; -\frac{2}{\beta} \sum_{(\mathrm{x}^+,\mathrm{x}^-)\in\mathcal{A}} \log\sigma\Big(\beta\big[r_{\boldsymbol{\theta}_{\mathrm{un}}}(\mathrm{x}^+) - r^g_{\boldsymbol{\theta}_{\mathrm{un}}}(\mathrm{x}^-)\big]\Big). \tag{25}$$

The normalization in Eq. 23 preserves the average scale of the dispreferred log-ratio, while shifting its token-level mass toward forget-specific positions.

NPO [Zhang et al., 2024]. NPO removes the preferred sequence from DPO and retains only the dispreferred log-ratio:

$$\mathcal{L}_{\mathrm{NPO}}(\boldsymbol{\theta}_{\mathrm{un}}\,;\,\mathcal{D}_{\mathrm{f}}) \;=\; -\frac{2}{\beta} \sum_{\mathrm{x}\in\mathcal{D}_{\mathrm{f}}} \log\sigma\big(-\beta\, r_{\boldsymbol{\theta}_{\mathrm{un}}}(\mathrm{x})\big). \tag{26}$$

The ATWU-augmented version replaces this log-ratio by its scorer-reweighted analog:

$$\mathcal{L}^g_{\mathrm{NPO}}(\boldsymbol{\theta}_{\mathrm{un}}\,;\,\mathcal{D}_{\mathrm{f}}) \;=\; -\frac{2}{\beta} \sum_{\mathrm{x}\in\mathcal{D}_{\mathrm{f}}} \log\sigma\big(-\beta\, r^g_{\boldsymbol{\theta}_{\mathrm{un}}}(\mathrm{x})\big). \tag{27}$$

As in DPO, the purpose is to preserve the sequence-level scale of the original objective while reallocating the forgetting signal across tokens.

SimNPO [Fan et al., 2025]. SIMNPO removes the reference model from NPO and uses a length-normalized, margin-shifted score. Let $\hat{\ell}(\boldsymbol{\theta}\,;\,\mathrm{x}) = \ell(\boldsymbol{\theta}\,;\,\mathrm{x})/|\mathrm{x}|$. The original loss is

$$\mathcal{L}_{\mathrm{SimNPO}}(\boldsymbol{\theta}_{\mathrm{un}}\,;\,\mathcal{D}_{\mathrm{f}}) \;=\; -\frac{2}{\beta}\sum_{\mathrm{x}\in\mathcal{D}_{\mathrm{f}}}\log\sigma\Big(\beta\big[\hat{\ell}(\boldsymbol{\theta}_{\mathrm{un}}\,;\,\mathrm{x}) - \delta\big]\Big). \tag{28}$$

We define the scored length-normalized loss as

$$\hat{\ell}^{g}(\boldsymbol{\theta}\,;\,\mathrm{x}) = \frac{\ell^{g}(\boldsymbol{\theta}\,;\,\mathrm{x})}{|\mathrm{x}|}.$$

The augmented objective is then

$$\mathcal{L}^{g}_{\mathrm{SimNPO}}(\boldsymbol{\theta}_{\mathrm{un}}\,;\,\mathcal{D}_{\mathrm{f}}) \;=\; -\frac{2}{\beta}\sum_{\mathrm{x}\in\mathcal{D}_{\mathrm{f}}}\log\sigma\Big(\beta\big[\hat{\ell}^{g}(\boldsymbol{\theta}_{\mathrm{un}}\,;\,\mathrm{x}) - \delta\big]\Big). \tag{29}$$

This keeps the same per-token averaging structure as SIMNPO, but makes the average selective: high-scored tokens contribute more to the forget loss, and low-scored structural tokens contribute less.

D.5 Ablations

Note on the ablation runs. The ablations reported in this subsection (Tables 20 and 21) were produced under an earlier version of the codebase whose headline ATWU run differs from the configuration used for the final results in Table 9. The absolute numbers therefore deviate by roughly 14 and 12 percentage points on FQ and UQ, respectively, from the latest ATWU entry in that table, but the relative trends and the qualitative conclusions of each ablation are unchanged. The sensitivity analysis in Section D.6 uses the final configuration.

Scorer regularizers. The scorer-side objective in Eq. 7 contains two explicit regularization terms: an entropy penalty, scaled by λ_{ent}, which encourages scores to become close to binary, and a population penalty, scaled by λ_{pop}, which anchors the mean score on $\mathcal{D}_{\mathrm{f}}$ to the target frequency ρ. In practice, we also apply ℓ_2 weight decay to the scorer parameters, scaled by λ_{w}. To isolate the contribution of each term, we toggle each regularizer between off ($\lambda = 0$) and on, where the on values are the values used in these runs, $\lambda_{\mathrm{ent}} = 1$, $\lambda_{\mathrm{pop}} = 10$, and $\lambda_{\mathrm{w}} = 1$. This gives the $2^3 = 8$-cell grid in Table 20, with all other components being constant.

Result. The population penalty is the dominant factor. Every configuration with λ_{pop} enabled achieves FQ ≥ 65, whereas every configuration without it remains in the narrow range FQ $\in [49, 52]$, regardless of the entropy or ℓ_2 setting. Thus, the mean-anchor is what allows the scorer to move away from the near-uniform initialization and commit to a sparse forget-specific subset.

Regularizers			**Relative**			**Utility**		
λ_{ent}	λ_{pop}	λ_{w}	FQ ↑	RD ↓	UQ ↑	MMLU	Rep.	WR
ORIGINAL			0.0	0.0	0.0	45.1	559	50.0
			49.8	**0.0**	49.8	45.1	565	49.0
		✓	51.4	1.4	50.0	45.0	568	53.0
	✓		<u>67.3</u>	3.3	63.9	45.0	570	48.0
	✓	✓	66.7	1.3	<u>65.4</u>	45.2	571	50.0
✓			49.3	<u>0.5</u>	48.7	45.1	568	53.5
✓		✓	50.3	5.0	45.3	45.2	567	51.0
✓	✓		65.1	3.9	61.2	45.1	574	50.0
✓	✓	✓	**70.5**	3.9	**66.6**	45.0	575	51.0

Table 20: Scorer regularizer ablation on TOFU `forget10`, `Llama-3.2-1B-Instruct`. ✓ indicates that the corresponding regularizer is on with the value used in these runs ($\lambda_{\mathrm{ent}} = 1$, $\lambda_{\mathrm{pop}} = 10$, $\lambda_{\mathrm{w}} = 1$); a blank cell means the regularizer is set to 0. Rows are ordered by binary expansion of $(\lambda_{\mathrm{ent}}, \lambda_{\mathrm{pop}}, \lambda_{\mathrm{w}})$ from all-off to all-on. The bottom row is the full regularized configuration used in this ablation.

Method	Relative			Utility		
	FQ ↑	RD ↓	UQ ↑	MMLU	Rep.	WR
ORIGINAL	0.0	0.0	0.0	45.1	559	50.0
PF	35.0	9.0	26.1	45.2	566	53.0
PU	33.4	8.7	24.7	45.3	566	51.0
TF	**73.7**	**2.1**	**71.6**	45.1	567	54.0
ATWU	70.5	3.9	66.6	45.0	575	51.0

(a) Scorer-state ablations. TF = TRAINED-FROZEN, PF = PRETRAIN-FROZEN, PU = PRETRAIN-UNFROZEN.

n_s	Relative			Utility		
	FQ ↑	RD ↓	UQ ↑	MMLU	Rep.	WR
ORIGINAL	0.0	0.0	0.0	45.1	559	50.0
1	31.3	**0.2**	31.1	45.0	565	53.0
5	**70.5**	3.9	**66.6**	45.0	575	51.0
10	67.0	3.6	63.4	45.2	578	49.0
joint	39.9	4.5	35.4	45.1	571	51.0

(b) Update-frequency sweep.

Table 21: Scorer training-procedure ablations on TOFU `forget10`, `Llama-3.2-1B-Instruct`. (a) Three alternatives to joint online training of the scorer. (b) Update-frequency sweep: the scorer is refreshed once per n_s model steps; $n_s = 5$ matches the headline configuration of ATWU.

The entropy and ℓ_2 terms have smaller, but still measurable, effects. With the population term enabled, removing the entropy penalty reduces FQ from 70.5 to 66.7, but improves RD from 3.9 to 1.3, for a modest UQ drop from 66.6 to 65.4. By contrast, removing ℓ_2 weight decay is more costly: FQ falls from 70.5 to 65.1 and UQ from 66.6 to 61.2, with essentially unchanged RD. The full configuration $(\lambda_{\text{ent}}, \lambda_{\text{pop}}, \lambda_{\text{w}}) = (1, 10, 1)$ achieves the best FQ and UQ overall, while remaining competitive on the other metrics. The only exception is RD, where the best values occur in under-committed $\lambda_{\text{pop}} = 0$ configurations that also fail to forget effectively. We therefore use the full regularizer set in the headline ATWU configuration.

Scorer training procedure. The preceding ablations fix the scorer training routine and vary either how the scorer is used or which regularizers shape it. We next ablate the training procedure itself, asking two questions: (i) whether the scorer must co-adapt with the language model during unlearning, and (ii) how often the scorer should be refreshed relative to model updates. Table 21 reports side-by-side ablations on TOFU `forget10` with `Llama-3.2-1B-Instruct`. All non-scorer settings match the headline ATWU run.

Table 21a compares three alternatives to the headline online procedure. TRAINED-FROZEN (TF) reuses the converged scorer from a previous ATWU run, freezes it, and then retrains the language model from scratch under this fixed weighting. PRETRAIN-FROZEN (PF) trains a fresh scorer for one epoch on the original, non-unlearned checkpoint and freezes it during unlearning. PRETRAIN-UNFROZEN (PU) uses the same pretrained scorer as an initialization, but continues to update it during unlearning.

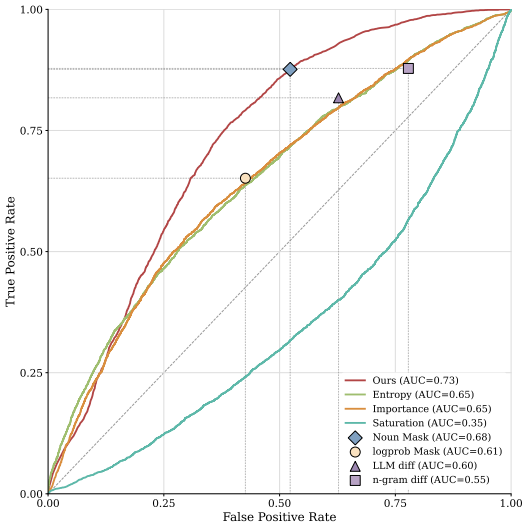


Figure 11: ROC curves for token-level forget-specificity detection on TOFU `forget10`. ATWU obtains the highest AUROC among the compared scoring methods; the dashed diagonal denotes random scoring.

The results show that online co-adaptation is not strictly necessary once a useful scorer has already been learned. TF slightly exceeds the headline online run on all relative metrics, improving FQ from 70.5 to 73.7, UQ from 66.6 to 71.6, and RD from 3.9 to 2.1. Thus, a converged scorer can be frozen and reused successfully. However, this does not hold for scorers trained only on the original model: PF reaches only FQ 35.0, and continuing to fine-tune that scorer in PU performs similarly poorly, with FQ 33.4. The key requirement is therefore not continuous co-training per se, but exposure to an unlearning trajectory. A scorer pretrained only on the original checkpoint is not a sufficient substitute.

Table 21b ablates the scorer update frequency. The scorer is refreshed once every $n_s \in \{1, 5, 10\}$ model steps, with all other settings fixed. Although updating every step is the most frequent option, it performs poorly: at $n_s = 1$, FQ drops to 31.3 and RD to 0.2, indicating that the method fails to move the model decisively in either the forget or retain direction. The headline setting $n_s = 5$ gives the best trade-off, achieving FQ 70.5 and UQ 66.6. Slowing the refresh rate to $n_s = 10$ remains competitive, with FQ 67.0 and UQ 63.4 at similar RD. This suggests a broad plateau between five and ten model steps: the exact refresh period is not especially sensitive, but updating the scorer every step is clearly detrimental.

Finally, the `joint` variant, which co-updates the scorer and language model in lockstep rather than using a scheduled refresh, also underperforms, reaching only FQ 39.9 and UQ 35.4. This confirms that the scheduled refresh mechanism, rather than mere simultaneous training, is what allows the scorer to commit to a useful forget-specific subset.

D.6 Sensitivity to the Budget ρ and Saturation β

We vary the two ATWU hyperparameters that most directly shape the forget update: the budget ρ, which sets the fraction of forget tokens the scorer is encouraged to select (Section 4), and the saturation parameter β of the forget loss (Eq. 6). Starting from the selected configuration on TOFU `forget10` with `Llama-3.2-1B-Instruct` ($\rho = 0.2$, $\beta = 7$; Table 6), we sweep $\rho \in \{0.1, 0.2, \ldots, 1.0\}$ at $\beta = 7$ and $\beta \in \{0, 1, \ldots, 10\}$ at $\rho = 0.2$, with all other settings fixed and a single seed. Since the selected configuration belongs to both sweeps, this amounts to 20 distinct runs. Because running the judge-based evaluation for every point would be costly, we report the forget and retain extraction strengths $\mathrm{ES}_{\mathcal{F}}$ and $\mathrm{ES}_{\mathcal{R}}$, MMLU, and Rep. We additionally report the share of forget tokens that the scorer scores above 0.5, averaged over the last 20% of scorer updates, which shows whether the budget constraint is active. Figure 12 shows the results, together with the original and retain models as reference points.

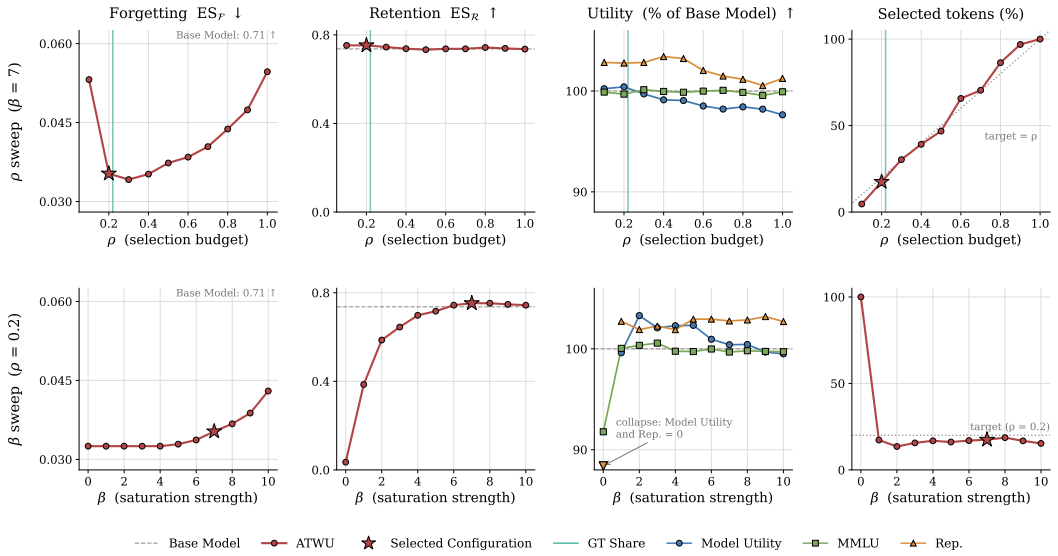


Figure 12: Sensitivity of ATWU to the budget ρ (top row, $\beta = 7$) and the saturation parameter β (bottom row, $\rho = 0.2$) on TOFU `forget10`, `Llama-3.2-1B-Instruct`, seed 0. Stars mark the selected configuration. Dashed and dotted lines mark the original model and the retain model, respectively; the original model's $\mathrm{ES}_{\mathcal{F}}$ (0.706) lies outside the plotted range. In the top row, the vertical line marks the ground-truth ratio of forget-specific tokens (0.22) in the labels of Zhou et al. [2026]. In the last column, the thin dashed line marks the target share of selected tokens (100ρ in the top row and 20% in the bottom row).

For β, results are stable over a broad plateau from about 6 to 10 ($\mathrm{ES}_{\mathcal{F}}$ 0.034–0.043, $\mathrm{ES}_{\mathcal{R}}$ 0.744–0.753). Setting $\beta = 0$ removes saturation and destroys retention ($\mathrm{ES}_{\mathcal{R}}$ 0.035, Rep. 0, MMLU 41.4), while small positive values keep utility intact but retain less ($\mathrm{ES}_{\mathcal{R}}$ 0.39 at $\beta = 1$ and 0.70 at $\beta = 4$). For ρ, values between 0.2 and 0.5 give similar forgetting ($\mathrm{ES}_{\mathcal{F}}$ 0.034–0.037), while both $\rho = 0.1$

and $\rho = 1.0$ forget less ($\mathrm{ES}_{\mathcal{F}}$ 0.053 and 0.055); every value of ρ still reaches a lower $\mathrm{ES}_{\mathcal{F}}$ than the retain model (0.059). MMLU and Rep. stay at or near their original values for every ρ. The realized selection share follows ρ, confirming that the budget constraint is active, and the best-performing $\rho = 0.2$ is close to the ground-truth ratio of forget-specific tokens in the labels of Zhou et al. [2026] (0.22).

E Additional Analysis of the Joint Optimization Formulation

E.1 Measuring Retain Conflict on TOFU

Whether the lowest-conflict subsets coincide with the semantically forget-specific tokens cannot be verified exhaustively, since each evaluation of κ is a full unlearning run, and there are exponentially many candidate subsets. We therefore estimate κ for several token sets of equal size on TOFU `forget10` with `Llama-3.2-1B-Instruct`.

We use the forget-specific token annotations from Zhou et al. [2026] as the ground-truth reference. We compare this labeled set against two baselines of the same size (3107 tokens): a random subset of all answer tokens and a random subset restricted to tokens not labeled forget-specific. Each estimate follows the definition of κ. Starting from the base model, before any TOFU fine-tuning, we train for 50 epochs on the retain split together with the forget loss restricted to the selected token set. We estimate $R^\star$ using matched control runs that optimize only the retain loss with the same seeds. To ensure that $\ell_{\mathrm{F}}^{\min}$ is finite, we use the clamped negative cross-entropy $-\min(\mathrm{CE}, 5)$ as the forget loss.

Table 22 reports the resulting estimates. The labeled forget-specific tokens have consistently lower estimated retain conflict than equally sized random subsets, whether the random tokens are drawn from all answer tokens or only from the structural-token pool. This provides empirical support for the premise that semantically forget-specific tokens are easier to suppress while preserving retain performance.

Table 22: Estimated retain conflict $\hat{\kappa}$ per retain token ($\times 10^{-2}$) on TOFU forget10 with `Llama-3.2-1B-Instruct`, for token sets of equal size (3107 tokens). Mean $\pm$ standard deviation over seeds, which vary the training seed and, for the random sets, the token draw.

Token set	seeds	$\hat{\kappa}$ ($\times 10^{-2}$)
Ground-truth forget-specific [Zhou et al., 2026]	6	7.1 ± 0.4
Ground-truth structural (random subset)	11	8.8 ± 0.5
Random	5	10.3 ± 0.5

These values should be interpreted comparatively rather than as accurate estimates of the absolute retain conflict. The reported values are differences relative to the matched control runs, whose retain loss after 50 epochs from the base model is $\hat{R}^\star = 1.44$ per retain token. For example, the optimization run using the labeled token set reaches a retain loss of 1.51 while driving the forget term to zero, yielding the estimated conflict reported above.

Because each value is obtained from a finite optimization run, it is only an upper bound on the corresponding κ. The released `retain90` checkpoint from TOFU illustrates that these bounds can be loose. Although this model is trained only on the retain split, it reaches a retain loss of 0.13, and its forget term on the labeled set is 0.06, giving a total objective of 0.19. This is far below the 1.51 reached by our finite-budget optimization and even below our estimate $\hat{R}^\star = 1.44$. Thus, within our computational budget, optimization of $\boldsymbol{\theta}$ is a more substantial limitation than the choice of token subset itself. For this reason, the estimates in Table 22 are meaningful primarily when compared under the same optimization budget.

E.2 Proof of Lemma 4.1 (Formal Statement)

Lemma E.1 (Formal Statement)**.** *Assume ρN_F is an integer and $|\ell_{\mathrm{f}}(\mathrm{x}_t \mid \mathrm{x}_{<t}; \boldsymbol{\theta})| \le M$ for all $\boldsymbol{\theta}, \mathrm{x}, t$ and some finite $M > 0$. There exists $\overline{\lambda}_{\mathrm{pop}} < \infty$ such that, for any $\lambda_{\mathrm{pop}} > \overline{\lambda}_{\mathrm{pop}}$, there exists $\overline{\lambda}_{\mathrm{ent}}(\lambda_{\mathrm{pop}}) < \infty$ such that, for any $\lambda_{\mathrm{ent}} > \overline{\lambda}_{\mathrm{ent}}(\lambda_{\mathrm{pop}})$, every global minimizer $(\boldsymbol{\theta}^\dagger, \mathbf{z}^\dagger)$ of*

$$\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta},\ \mathbf{z} \in [0,1]^{N_F}} \widetilde{\mathcal{L}}(\boldsymbol{\theta}, \mathbf{z})$$

satisfies $\mathbf{z}^\dagger \in \{0,1\}^{N_F}$ with $\sum_{\mathrm{x},t} z^\dagger_{\mathrm{x},t} = \rho N_F$, and is itself a global minimizer of (3).

Proof. Let $V_c^\star := \inf_{\boldsymbol{\theta},\, \mathbf{z} \in \mathcal{Z}} \mathcal{L}(\boldsymbol{\theta}, \mathbf{z})$. We argue in two steps: λ_{ent} forces $\mathbf{z}$ to be binary, and λ_{pop} then forces the budget to be tight.

Step 1 (binarity). Let $(\boldsymbol{\theta}', \mathbf{z}')$ minimize $\widetilde{\mathcal{L}}$ over $\boldsymbol{\Theta} \times [0,1]^{N_F}$, and suppose some coordinate $z'_{\mathrm{y},k} \in (0,1)$. Fix all other coordinates and round $z'_{\mathrm{y},k}$ to its closest endpoint $b \in \{0,1\}$, denoting the result by $\mathbf{z}''$. Write $\eta \coloneqq \min(z'_{\mathrm{y},k}, 1 - z'_{\mathrm{y},k}) \in (0, \frac{1}{2}]$.

The smooth part $\mathcal{L} + \lambda_{\mathrm{pop}}(\cdot)^2$ is Lipschitz in $z_{\mathrm{y},k}$ with constant at most $M + 2\lambda_{\mathrm{pop}}/N_F$, so the absolute change in the smooth part under rounding is at most $(M + 2\lambda_{\mathrm{pop}}/N_F)\eta$. The entropy contribution satisfies $H(z'_{\mathrm{y},k}) \geq c_0\eta$ for $c_0 \coloneqq \log 2$, and it becomes zero after rounding. Thus, if

$$\lambda_{\mathrm{ent}} > \overline{\lambda}_{\mathrm{ent}}(\lambda_{\mathrm{pop}}) \coloneqq \tfrac{1}{c_0}\big(M + 2\lambda_{\mathrm{pop}}/N_F\big),$$

then $\widetilde{\mathcal{L}}(\boldsymbol{\theta}', \mathbf{z}'') < \widetilde{\mathcal{L}}(\boldsymbol{\theta}', \mathbf{z}')$, contradicting optimality. Hence $\mathbf{z}' \in \{0,1\}^{N_F}$.

Step 2 (budget). With $\mathbf{z}' \in \{0,1\}^{N_F}$, the entropy term vanishes. Suppose $\sum_{\mathrm{x},t} z'_{\mathrm{x},t} \neq \rho N_F$; since both quantities are integers, $|\sum z'_{\mathrm{x},t} - \rho N_F| \geq 1$, so the budget term is at least $\lambda_{\mathrm{pop}}/N_F^2$. For any feasible $(\boldsymbol{\theta}^\star, \mathbf{z}^\star) \in \boldsymbol{\Theta} \times \mathcal{Z}$, both regularizer terms vanish at $(\boldsymbol{\theta}^\star, \mathbf{z}^\star)$, so

$$\mathcal{L}(\boldsymbol{\theta}', \mathbf{z}') + \tfrac{\lambda_{\mathrm{pop}}}{N_F^2} \;\leq\; \widetilde{\mathcal{L}}(\boldsymbol{\theta}', \mathbf{z}') \;\leq\; \widetilde{\mathcal{L}}(\boldsymbol{\theta}^\star, \mathbf{z}^\star) \;=\; \mathcal{L}(\boldsymbol{\theta}^\star, \mathbf{z}^\star).$$

Taking the infimum over $(\boldsymbol{\theta}^\star, \mathbf{z}^\star) \in \boldsymbol{\Theta} \times \mathcal{Z}$ yields $\mathcal{L}(\boldsymbol{\theta}', \mathbf{z}') + \frac{\lambda_{\mathrm{pop}}}{N_F^2} \leq V_c^\star$. Since $R(\boldsymbol{\theta}) \geq 0$ and $\ell_{\mathrm{F}} \geq -M$, $\mathcal{L}(\boldsymbol{\theta}, \mathbf{z}) \geq -MN_F$ on $\boldsymbol{\Theta} \times [0,1]^{N_F}$. Setting $\overline{\lambda}_{\mathrm{pop}} \coloneqq N_F^2(V_c^\star + MN_F)$, any $\lambda_{\mathrm{pop}} > \overline{\lambda}_{\mathrm{pop}}$ would imply $\mathcal{L}(\boldsymbol{\theta}', \mathbf{z}') < -MN_F$, contradicting this lower bound. Hence $\sum z'_{\mathrm{x},t} = \rho N_F$.

Step 3 (optimality for the constrained problem). For $(\boldsymbol{\theta}', \mathbf{z}') \in \boldsymbol{\Theta} \times \mathcal{Z}$, $\widetilde{\mathcal{L}}(\boldsymbol{\theta}', \mathbf{z}') = \mathcal{L}(\boldsymbol{\theta}', \mathbf{z}')$, and the same holds for any feasible competitor. Hence $\mathcal{L}(\boldsymbol{\theta}', \mathbf{z}') \leq \mathcal{L}(\boldsymbol{\theta}^\star, \mathbf{z}^\star)$ for all $(\boldsymbol{\theta}^\star, \mathbf{z}^\star) \in \boldsymbol{\Theta} \times \mathcal{Z}$, i.e. $(\boldsymbol{\theta}', \mathbf{z}')$ is a global minimizer of (3). □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the algorithmic, theoretical, and empirical contributions, and the experiments in Section 5 support those claims within the stated benchmark and model scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations section discussing small forget sets, assumptions behind the theory, benchmark/model coverage, single-run final evaluations, and possible judge-model dependence.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The problem setting and method sections state the assumptions for the theoretical results, and the appendix provides the formal proofs and supporting derivations.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5 and Sections B and C specify the benchmarks, base models, metrics, tuning protocol, hyperparameter ranges and selected values, and final-run settings to reproduce the reported results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper cites and links the existing public benchmarks and codebases it builds on, and the supplement provides prompts and selected hyperparameters. A standalone anonymized reproduction package with runnable scripts is not yet included but will be upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer:[Yes]

   Justification: Sections C and 5 describe the data splits, base models, hyperparameter-selection procedure, search ranges, selected hyperparameters, optimizers, schedules, precision, batch sizes, and evaluation protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports mean and standard deviation for the token-ranking diagnostic, but the main unlearning tables are single final-evaluation runs due to compute and judge-model cost and therefore do not include across-run error bars or significance tests.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section C reports GPU type, memory, number of GPUs, training epochs, batch sizes, precision, optimizers, and judge-LLM API cost.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The work uses existing public benchmarks and models, studies machine unlearning as a privacy-preserving intervention, and does not introduce a new deployed system, scraped dataset, or high-risk model release.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Although there is no dedicated broader impact section, the paper introduction and limitations motivate unlearning through privacy and safety.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: Paper poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: The paper credits and links the datasets, base models, and codebases used in the experiments, but does not explicitly list the license or terms of use for each asset.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce a new dataset, benchmark, pretrained model, or other standalone asset; the supplementary prompts and hyperparameter files document the evaluation setup.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: Paper does not involve crowdsourcing nor research with human subjects

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Sections B.2, B.3 and C describe the use of an LLM judge for semantic forget/retain evaluation and win-rate evaluation, including prompts in the supplementary material and validation against human annotations.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.