NOT EVERY CHANGE IS NECESSARY: RECOVERABLE DRIFT IN LARGE LANGUAGE MODEL UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning in large language models aims to remove unwanted knowledge while preserving the model's remaining capabilities. Although existing methods use retention objectives or restrict where edits occur, achieving the desired forgetting level can still leave collateral changes that impair non-target behavior. Our recovery comparisons suggest that some of these changes can be reversed while preserving observed forgetting performance. In this work, we present **P**ropose-**T**hen-**P**roject Unlearning (PTP-U), a framework that combines targeted forgetting with the recovery of non-target capabilities. PTP-U first applies local analytic edits to weaken target knowledge associations, then aligns non-target output distributions with those of the original model to recover capabilities while maintaining fixed forgetting constraints. Both stages serve a common goal: satisfying the forgetting requirements while preserving fluent generation and performance on non-target tasks. Across three benchmarks, PTP-U achieves the strongest forgetting–retention trade-off among evaluated methods, reaching 81.22%–91.03% forgetting while preserving 94.20% non-target utility on average. At matched forgetting, PTP-U consistently retains higher non-target utility.

1 INTRODUCTION

Large language models (LLMs) acquire extensive knowledge, skills, and linguistic patterns during pre-training (Pan et al., 2026; Guo et al., 2025a; Ranjan et al., 2026), becoming general-purpose models for long-term deployment (Zhao et al., 2025; Ye et al., 2026). However, training corpora may include private information (Ramakrishna et al., 2025), copyrighted content (Shi et al., 2025a), or unsafe knowledge (Li et al., 2024), requiring the removal of specific information after training (Yang et al., 2026). While retraining from scratch after excluding target data can achieve this goal, it is computationally infeasible for modern LLMs (Gong et al., 2026; Ye et al., 2025). Therefore, LLM unlearning aims to selectively remove target knowledge while preserving the model's original capabilities and behaviors on non-target inputs.

Existing unlearning methods mainly apply forgetting signals at the output level by suppressing target content generation (Pu et al., 2026), adjusting output distributions (Ma et al., 2026), or optimizing preference objectives (Fan et al., 2025). They are often combined with retain objectives or regularization to preserve non-target behaviors. However, output-level objectives offer limited control over which parameters change and by how much, making it difficult to confine updates to parameters associated with target knowledge (Wang et al., 2025b). Another line of methods directly modifies internal representations or local parameters to weaken target knowledge associations. These methods control modification scopes through low-rank updates (Chen et al., 2026), local constraints (Shen et al., 2026), or analytical editing (Li et al., 2026b). Although internal intervention more directly affects computations related to target knowledge, local changes still propagate nonlinearly through deep Transformer architectures, making global behavioral changes difficult to fully control.

Therefore, achieving the desired forgetting level does not imply that all accompanying changes are necessary. Some changes may be redundant perturbations from optimization, disrupting the model's semantic structure and causing semantic drift or capability degradation (Wang et al., 2025a; Lee et al., 2026). **This motivates a key question: can such collateral changes be reversed while preserving forgetting?** To explore this, we investigate recoverable changes in models that have achieved the target forgetting level. We adjust the unlearned model toward the Base model while re-

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

taining the forgetting effect. As shown in Figure 1, if certain changes can be restored without recovering the forgotten knowledge, they are unnecessary for unlearning and instead reflect optimization-induced drift. We define such reversible changes as *recoverable drift*.

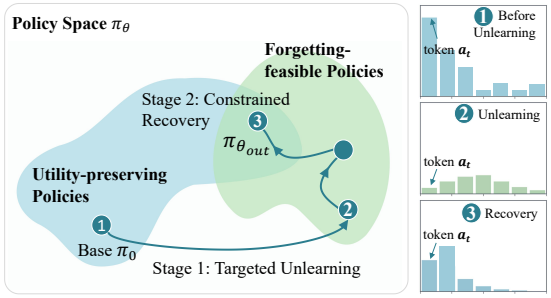


Figure 1: PTP-U first achieves forgetting feasibility (Stage I), then restores utility while maintaining feasibility (Stage II). Right: target-answer suppression persists during recovery.

Prior post-training studies show that among models solving the same new task, solutions with output behaviors closer to the Base model retain more original capabilities (Shenfeld et al., 2026). Inspired by this foundation, we propose Propose-Then-Project Unlearning (PTP-U), which combines local analytical editing with constrained recovery fine-tuning. PTP-U first constructs candidate models through local analytic edits and then reduces behavioral drift by aligning non-target outputs with the Base model under fixed forgetting constraints.

Specifically, PTP-U addresses the same constrained objective at two scales. In the local proposal stage, we use Fisher geometry to approximate Base-policy drift quadratically and efficiently generate candidate updates under linearized forgetting constraints. When updates move beyond the local region, Transformer nonlinearities weaken this approximation. Thus, the nonlinear policy projection stage operates directly on output distributions to reduce deviation from the Base policy under fixed forgetting constraints. For sequential Transformer modules, PTP-U re-estimates the local geometry and constraints of downstream components after upstream updates, reducing linearization errors. This separates local proposal generation from subsequent nonlinear recovery while retaining a common objective and fixed forgetting requirements. Across three benchmarks, PTP-U achieves the strongest forgetting/retention trade-off, reaching 81.22%–91.03% forgetting (compared with 80.70%–87.93% for the strongest forgetting baseline) while preserving the highest average capability retention of 94.20%. At matched forgetting, it consistently retains higher utility (93.32%–95.86%) and remains closer to the Base model. These results support our central finding that a substantial portion of the changes induced by strong unlearning is not necessary for maintaining forgetting. In summary, our contributions are:

(1) We propose Propose-Then-Project Unlearning (PTP-U), a Base-centered framework that couples targeted forgetting with explicit behavioral recovery. It separates forgetting-oriented proposal generation from recovery toward the Base policy under fixed forgetting constraints.

(2) We develop an analytic editing mechanism that constructs local proposals under linearized forgetting constraints. Sequential relinearization updates downstream geometry and constraints after upstream edits, accounting for dependencies within Transformer blocks.

(3) We validate PTP-U across multiple benchmarks and model families, demonstrating strong forgetting–retention trade-offs, robustness against prompt-based recovery and relearning attacks, and stable performance across diverse settings.

2 Recoverable Drift: From Observation to Formulation

2.1 Forgetting Constraints and Behavioral Drift

Let $\pi_0 = \pi_{\theta_0}$ be the Base model, π_θ the updated model, and $\mathcal{U}_f$ the set of target knowledge units. For each $u \in \mathcal{U}_f$, we measure relation support using a target query q, a matched contrastive query $\bar{q}$ without the target relation, and an answer a, where $|a|$ denotes the number of tokens in a:

$$S_u(\theta) = \mathbb{E}_{(q,\bar{q},a)\sim\mathcal{V}_u}\left[\ell_\theta(a \mid q) - \ell_\theta(a \mid \bar{q})\right], \qquad \ell_\theta(a \mid q) = \frac{1}{|a|}\sum_{t=1}^{|a|}\log \pi_\theta(a_t \mid q, a_{<t}), \quad (1)$$

where $\mathcal{V}_u$ contains query views and answer aliases. Lower scores $S_u(\theta)$ indicate weaker relation support. We fix thresholds b_u before updating and define the forgetting-feasible set $\mathcal{C}_f$, which contains models satisfying all target-unit thresholds:

$$\mathcal{C}_f = \{\theta \mid c_u(\theta) \le 0,\ \forall u \in \mathcal{U}_f\}, \qquad c_u(\theta) = S_u(\theta) - b_u. \quad (2)$$

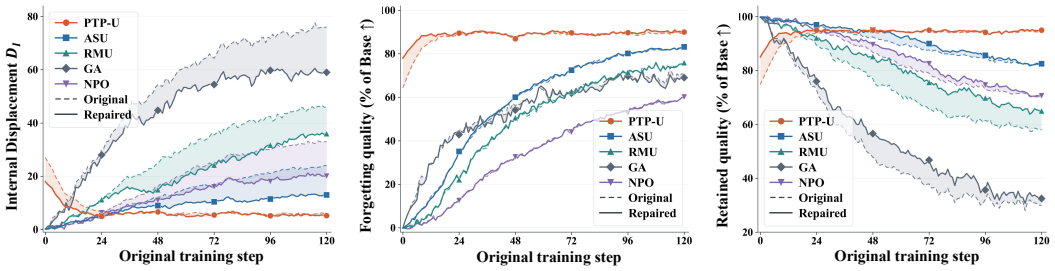


Figure 2: Checkpoint-wise recovery across unlearning methods. Dashed lines show original checkpoints and solid lines their independently repaired copies. Shading marks paired differences. The horizontal axis records fine-tuning steps. PTP-U step 0 follows the *Stage I* analytic edits. Repairs are performed separately and do not feed into subsequent original checkpoints.

Sample construction and threshold calibration are detailed in Appendices C.2 and C.7. On a fixed non-target context distribution μ_a, we define Base-policy drift as

$$D_{\mathrm{B}}(\theta) = \mathbb{E}_{s\sim\mu_a} D_{\mathrm{KL}}\left(\pi_0(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)\right). \tag{3}$$

Here, D_{KL} denotes the Kullback–Leibler (KL) divergence, and s includes the input and its generation prefix. Smaller $D_{\mathrm{B}}(\theta)$ indicates closer next-token distributions on these contexts.

2.2 Recoverable Drift and Constrained Unlearning Formulation

We examine whether changes introduced during unlearning remain necessary once forgetting has been achieved. Results from Figure 2 address this question through paired recovery comparisons. To track internal changes, let $h_\theta^\ell(s)$ denote the hidden representation at layer ℓ and P_{ret} the orthogonal projector onto a fixed retain-sensitive subspace. We focus on displacement within this subspace and normalize by the Base representation energy to express changes relative to the original representation scale. We measure internal displacement as

$$D_{\mathrm{I}}(\theta) = 100\sqrt{\frac{\mathbb{E}_{s\sim\mu_a}\left\|P_{\mathrm{ret}}\left(h_\theta^\ell(s) - h_{\theta_0}^\ell(s)\right)\right\|_2^2}{\mathbb{E}_{s\sim\mu_a}\left\|P_{\mathrm{ret}} h_{\theta_0}^\ell(s)\right\|_2^2 + \varepsilon}}, \tag{4}$$

where $\|\cdot\|_2^2$ denotes the squared L2 norm and $\varepsilon > 0$ is a fixed numerical stabilizer. The contexts, layer, and projector remain fixed across comparisons. Appendix C.4 specifies subspace construction, metric definitions, and repair settings.

Recovery generally reduces internal displacement while leaving forgetting quality largely unchanged, with retention gains varying across methods. These empirical results support our *recoverable drift* premise: some internal displacement introduced by unlearning can be reduced without undoing forgetting. Accordingly, model selection should consider both forgetting feasibility and behavioral deviation from the Base model. Inspired by this, we treat recovery as an explicit part of the unlearning pipeline, instead of treating all unlearning-induced modifications as indispensable.

While $D_{\mathrm{I}}(\theta)$ is a representation diagnostic, we use D_{B} to measure behavioral deviation directly in output space and formulate our constrained unlearning objective as

$$\underset{\theta\in\Theta}{\text{minimize}}\; D_{\mathrm{B}}(\theta) \qquad \text{s.t.} \quad S_u(\theta) \le b_u, \qquad \forall u \in \mathcal{U}_f, \tag{5}$$

where Θ denotes the model's parameter space. This objective motivates an explicit recovery phase while keeping the forgetting thresholds fixed throughout optimization. Our method first applies local analytic edits to seek feasible solutions, then performs constrained recovery to reduce undesirable behavioral drift of the model. Global optimality is not assumed.

3 Propose-Then-Project Unlearning (PTP-U)

PTP-U approximately solves Eq. 5 via two stages. It first generates a forgetting proposal from the local approximation of the current model and verifies a feasible initialization with the actual model. Then, nonlinear policy projection reduces Base-policy drift while maintaining forgetting constraints. Attention→FFN sequential re-linearization improves the accuracy of local updates in Transformers.

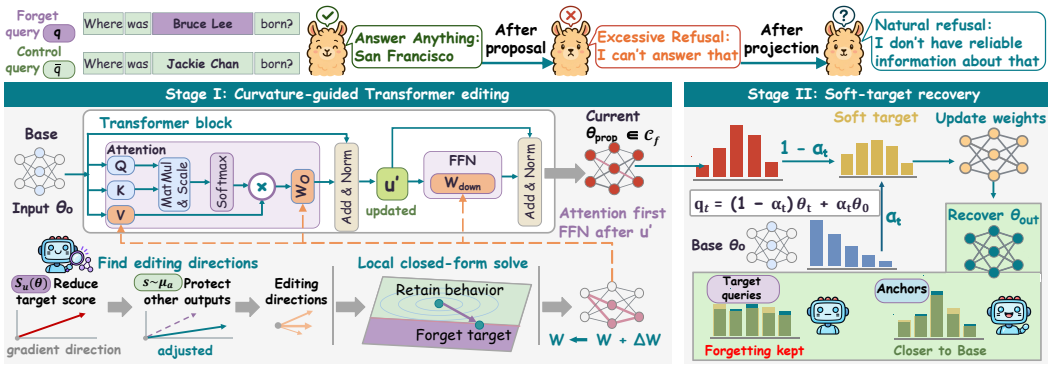


Figure 3: Overview of PTP-U. *Stage I* computes a curvature-guided closed-form proposal in an editable subspace. It updates Attention first, refreshes the operating point, then updates FFN to obtain a feasible unlearned model. *Stage II* constructs a soft target from current and Base policies for constrained recovery. It preserves forgetting while reducing behavioral drift and excessive refusal.

3.1 Local Analytic Proposal with Curvature Guidance

Starting from the Base parameters, we construct the update around the current model θ_k. To reduce computational cost, we restrict the parameter change to $\delta = U_k z$, where U_k is a full column rank low-dimensional editable basis and z is the optimized coefficient vector. The editing directions are derived from target knowledge gradients and preconditioned by the K-FAC (Martens & Grosse, 2015) approximation of the current model Fisher information (Amari, 1998) on fixed anchors, capturing the local influence of different directions on non-target outputs. The same curvature approximation defines the local quadratic model below. Let $c_k = c(\theta_k)$ denote the stacked constraint residuals and J_k the Jacobian with respect to z. We define the local model as

$$m_k(z) = \widetilde{g}_k^\top z + \frac{1}{2} z^\top H_k z, \qquad c(\theta_k + U_k z) \approx c_k + J_k z, \tag{6}$$

where m_k approximates the change of D_{B}, $\widetilde{g}_k = U_k^\top \nabla_\theta D_{\mathrm{B}}(\theta_k)$, and $H_k = U_k^\top (\widehat{F}_{a,k} + \lambda I) U_k$ with $\lambda > 0$. Here, $F_{a,k}$ is the Fisher information of the current model on fixed anchors, and $\widehat{F}_{a,k}$ is its positive semidefinite K-FAC approximation. For the next-token softmax distribution, $F_{a,k}$ corresponds to the generalized Gauss-Newton curvature with logits as output variables. The damping term improves stability and, together with the full column rank of U_k, ensures $H_k \succ 0$.

The Base gradient vanishes, but $\widetilde{g}_k$ is generally nonzero at later iterates and is retained in m_k. Appendix 1 derives the gradient and separates Fisher curvature from the additional logit-curvature term. Appendix 3 quantifies the error of the damped surrogate. To cap the required residual reduction in each iteration, we define the truncated constraint residual as

$$r_{u,k} = \min\{c_u(\theta_k), r_{\max}\}, \qquad r_{\max} > 0. \tag{7}$$

Here, $r_{\max}$ is a hyperparameter that caps the required reduction for unsatisfied constraints. For satisfied constraints, the negative residual is preserved as the boundary margin for linearized feasibility checking. We optimize the coefficients z by minimizing the local objective $m_k(z)$:

$$\underset{z}{\text{minimize}} \quad m_k(z) \qquad \text{subject to} \quad J_k z \le -r_k. \tag{8}$$

Since large positive residuals are truncated, a single iteration may only achieve partial forgetting. For a fixed active set A, let J_A and r_A denote the corresponding constraint matrix and residual. When J_A has full row rank, the unconstrained recovery direction v_k and constraint multipliers ν_A yield the analytical solution

$$v_k = -H_k^{-1}\widetilde{g}_k, \quad \nu_A = \left(J_A H_k^{-1} J_A^\top\right)^{-1} \left(J_A v_k + r_A\right), \quad z_k^{\mathrm{prop}} = v_k - H_k^{-1} J_A^\top \nu_A. \tag{9}$$

Here, $(\cdot)^{-1}$ denotes the matrix inverse. If $\nu_A \ge 0$ and all remaining inequalities hold, Eq. 9 is the unique solution to Eq. 8 by Appendix 2. The first term minimizes the unconstrained local model, and the second gives the minimum H_k-metric correction enforcing the active constraints. Since Eq. 9 only solves the local problem, the local model must be reconstructed after the update. This local solution does not provide a global closed-form solution to the nonlinear unlearning problem.

3.2 SEQUENTIAL RELINEARIZATION IN TRANSFORMERS

The local proposal considers the computation order of Transformer blocks, formulated as

$$u_l = h_l + A_l(h_l), \qquad h_{l+1} = u_l + F_l(u_l), \tag{10}$$

where $A_l(\cdot)$ and $F_l(\cdot)$ represent the Attention and FFN submodules, respectively. For a fixed block input, updating Attention parameters θ_A affects the FFN input through the following relation, where $J_{F_l}(u_l)$ denotes the FFN input Jacobian. The first-order joint model captures the propagation of Attention changes through FFN.

$$\frac{\partial h_{l+1}}{\partial \theta_A} = (I + J_{F_l}(u_l)) \frac{\partial A_l(h_l)}{\partial \theta_A}. \tag{11}$$

Therefore, we first solve δ_A in the current Attention editing subspace and update parameters as $\theta_k^A = \theta_k + \delta_A$. Then, we perform a forward pass to refresh the knowledge unit residuals and Base recovery gradients under θ_k^A, recompute FFN K-FAC curvature statistics on fixed anchors, and update the FFN editing basis with its constraint Jacobian to solve for δ_F. Relinearization at θ_k^A removes the stale-Jacobian term from the FFN error decomposition, as shown in Appendix 4.

Each candidate update is evaluated on the actual model to measure forgetting progress and constraint preservation. If the predicted change differs significantly from the actual change or violates previous constraints, the update is rejected, the target advancement magnitude is reduced, and optimization is repeated. This process continues until a feasible checkpoint $\theta_{\text{prop}} \in \mathcal{C}_f$ is obtained, which serves as the initialization for the next stage.

3.3 NONLINEAR POLICY PROJECTION

The finite-dimensional editing subspace and local curvature approximation constrain proposal accuracy, while larger updates may cause behavioral changes beyond the local model. Thus, the second stage starts from θ_{prop} and continues optimizing Eq. 5 on the true output distribution. At projection iteration t, let $\theta_t \in \mathcal{C}_f$ denote the current feasible model. We construct a progressive recovery target towards the Base model on fixed anchors:

$$q_t(\cdot \mid s) = (1 - \alpha_t)\pi_{\theta_t}(\cdot \mid s) + \alpha_t \pi_0(\cdot \mid s), \qquad 0 < \alpha_t \leq 1. \tag{12}$$

Here, q_t is fixed and gradients are stopped during the inner-loop optimization, while α_t determines the shift of the target distribution toward the Base model. At each round, we minimize $\mathcal{L}_t(\theta)$ over the forgetting feasible region $\theta \in \mathcal{C}_f$, where $\mathcal{L}_t(\theta)$ is defined as

$$\begin{aligned} \mathcal{L}_t(\theta) &= \mathbb{E}_{s\sim\mu_a} D_{\text{KL}}(q_t(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)) \\ &= \alpha_t D_{\text{B}}(\theta) + (1 - \alpha_t)\mathbb{E}_{s\sim\mu_a} D_{\text{KL}}(\pi_{\theta_t}(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)) + C_t, \end{aligned} \tag{13}$$

where C_t is independent of θ. Minimizing $\mathcal{L}_t$ over $\mathcal{C}_f$ defines a proximal step for Eq. 5, combining Base recovery with a penalty on policy changes. Appendix 5-6 establish the decomposition and sufficient descent conditions, respectively. The optimization is conducted by adaptively updating the Lagrangian multiplier to satisfy the unlearning constraint (Dhillon et al., 2025), and candidate solutions are verified on the real model. Only updates meeting the following conditions are accepted:

$$\theta_{t+1} \in \mathcal{C}_f, \qquad D_{\text{B}}(\theta_{t+1}) < D_{\text{B}}(\theta_t). \tag{14}$$

If verification fails, the update is reverted and the recovery magnitude is reduced. Base only provides the recovery distribution over non-target anchors. Since all inputs share the same model parameters, recovery must preserve the fixed forgetting constraint to avoid restoring target knowledge. When the computational budget is exhausted or no candidate satisfying Eq. 14 is found within the tolerance, the accepted feasible model θ_{out} is returned. Here, projection denotes the constrained recovery process in Eq. 5, which does not ensure the globally nearest point. The complete update, rollback, and stopping rules are described in Appendix C.8. The editing basis, curvature statistics, and recovery objective are used only during training and introduce no additional inference modules.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Datasets. We evaluate PTP-U on three benchmarks. RWKU evaluates the removal of real-world entity knowledge (Jin et al., 2024). WMDP measures hazardous knowledge in biosecurity (Bio) and cybersecurity (Cyber) (Li et al., 2024). MUSE-Books evaluates copyright unlearning on the Harry Potter corpus (Shi et al., 2025a). MMLU measures the preservation of general knowledge (Hendrycks et al., 2020). Dataset and split details are detailed in Appendix C.1.

Evaluation Metrics. For RWKU, we report ROUGE-L recall on fill-in-the-blank (FB), question-answering (QA) probes, and adversarial attack (AA) scenarios. For WMDP, we report accuracy on Bio/Cyber. The 25% random-choice indicates best forgetting. For MUSE-Books, we report BLEU and ROUGE-L. MMLU accuracy (Gen.) and GPT-4o fluency (Xu et al., 2025b) assess retained utility. We report R-Ent. separately on RWKU and WMDP to measure confirmed restricted knowledge re-Entry after an initial risk closure. Metric details are in Appendix C.3.

Baselines. We compare PTP-U with NPO_KL (Zhang et al., 2024), RMU (Li et al., 2024), WHP (Eldan & Russinovich, 2023), ALTER (Chen et al., 2026), Attention-Smoothing (ASU) (Zare Zade et al., 2026) and ICUL (Pawelczyk et al., 2024), adapted SCANS (Cao et al., 2025), MET (Yu et al., 2025). The details of Baselines are provided in Appendices C.5 and C.6.

Implementation Details. Across all backbones, we set $r_{\max} = 0.25$, $\lambda = 1.0 \times 10^{-3}$, and $\alpha_t = 0.5$ as the initial recovery mixing coefficient. We select the middle layers (23) of the model for editing. The forgetting boundaries $b_u = S_u(\theta_0) - \Delta_{\mathrm{f}}$, with $\Delta_{\mathrm{f}} = \log 2$. We use four NVIDIA RTX 4090 GPUs. Model-specific settings are provided in Appendix C.7.

4.2 MAIN RESULTS

Forget-retain trade-off. Table 1 evaluates long-form copyright continuation on Llama2-7B. PTP-U achieves the lowest BLEU and R-L while preserving the highest MMLU, indicating a strong balance between forgetting and retained capability. Table 2 further evaluates this trade-off in two restricted-knowledge settings. RWKU measures entity forgetting with FB and QA, and AA evaluates robustness to adversarial recovery. WMDP measures Bio/Cyber forgetting and retained general capability. On Llama3.1-8B-Instruct, PTP-U preserves 65.54 Gen. on RWKU and 63.80 on WMDP. ALTER and ASU achieve stronger suppression on several direct forgetting metrics, but cause larger utility degradation and weaker AA robustness on RWKU. Representation intervention and parameter editing are generally more robust to adversarial recovery, but often reduce general capability substantially. ICUL preserves fluency through inference-time prompting but provides weaker forgetting. Overall, PTP-U maintains a favorable trade-off. Its local analytic proposal satisfies forgetting constraints, while nonlinear policy projection reduces drift from the Base model without relaxing them. This separation enables recovery from behavioral changes caused by the proposal across unlearning tasks. Case analyses are in Appendix E.

Table 1: Copyright unlearning results.

Method	Forget Perf.		Retain Perf.	
	BLEU↓	R-L↓	MMLU↑	Flu.↑
Original	74.76	98.68	46.39	4.03
WHP	23.55	17.93	38.49	2.52
ICUL	38.47	32.49	39.86	**3.60**
SCANS	19.92	17.72	<u>42.05</u>	3.27
ALTER	12.96	10.40	41.84	3.08
ASU	<u>10.72</u>	<u>9.68</u>	37.56	3.28
PTP-U	**8.61**	**6.33**	**43.34**	<u>3.34</u>

Hyperparameter sensitivity. In figure 4, we examine $\mathcal{L}_{\mathrm{edit}}$, λ, and $r_{\max}$ by varying one factor while fixing the others at the reference configuration. The layer axis denotes the final layer of a fixed three-layer editing window. The favorable trade-off near layer 23 suggests that this middle-to-late window provides a more selective intervention point for knowledge-dependent computations, rather than a general advantage of deeper editing. The other hyperparameters control Stage I. Specifically, λ regularizes the local curvature metric, while $r_{\max}$ limits constraint-residual reduction per analytic iteration. Across the tested range, increasing λ generally improves retention at the expense of forgetting. Increasing $r_{\max}$ from 0.05 to 1.00 reduces WMDP accuracy from 40.68% to 25.92%, closer to chance, while decreasing MMLU from 66.84% to 58.84%, indicating greater collateral damage under aggressive target advancement. As $r_{\max} \to 0$, the required positive progress on violated constraints vanishes. As $\lambda \to 0$, curvature damping disappears and numerical stability may degrade. Neither limit necessarily disables the full pipeline. We therefore use the window ending

Table 2: Main results on RWKU and WMDP. †, ‡, and ⋆ denote output-distribution optimization, representation intervention & parameter editing, and prompt-based unlearning methods, respectively.

Baseline	RWKU					WMDP		
	FB↓	QA↓	AA↓	Gen.↑	Flu.↑	Bio.↓	Cyber↓	Gen.↑
Llama3.1-8B-Instruct (Grattafiori et al., 2024)								
Base	63.95	66.42	69.83	68.37	4.20	72.74	47.35	68.37
ICUL⋆	43.23	32.57	42.60	64.80	**4.07**	45.90	33.34	61.10
RMU‡	19.10	23.84	21.30	54.71	2.96	32.52	32.75	53.43
MET‡	33.64	28.47	23.54	58.52	3.18	34.63	32.47	52.19
SCANS‡	16.80	23.47	21.05	60.40	3.61	31.70	34.00	58.20
NPO_{KL}†	36.02	29.10	26.89	60.29	3.30	38.38	35.32	58.37
ALTER†	18.61	22.51	22.95	61.38	3.20	29.69	31.77	61.60
ASU†	**8.90**	**10.62**	26.80	60.89	3.43	**27.45**	32.48	57.74
PTP-U†‡	11.42	13.08	**15.20**	**65.54**	3.93	28.14	**29.84**	**63.80**
Qwen3-14B (Yang et al., 2025)								
Base	63.06	45.42	56.02	79.35	4.41	75.51	61.06	79.35
ICUL⋆	37.23	31.57	32.60	69.83	**4.22**	46.82	31.50	68.22
RMU‡	22.15	25.71	**14.33**	62.30	2.96	43.85	36.24	66.51
SCANS‡	21.50	28.11	26.40	68.70	3.79	34.60	35.80	67.50
ALTER†	20.60	23.47	27.87	71.02	3.35	34.24	**30.58**	70.55
ASU†	**11.30**	15.10	24.94	69.53	3.43	**29.40**	31.58	68.38
PTP-U†‡	13.76	**14.55**	15.84	**72.62**	3.90	30.14	31.23	**72.42**

at layer 23, $\lambda = 10^{-3}$, and $r_{\max} = 0.25$, yielding 28.99% WMDP and 63.80% MMLU accuracy. Additional Stage II analyses and internal visualizations are provided in Appendix D.2 and E.1.

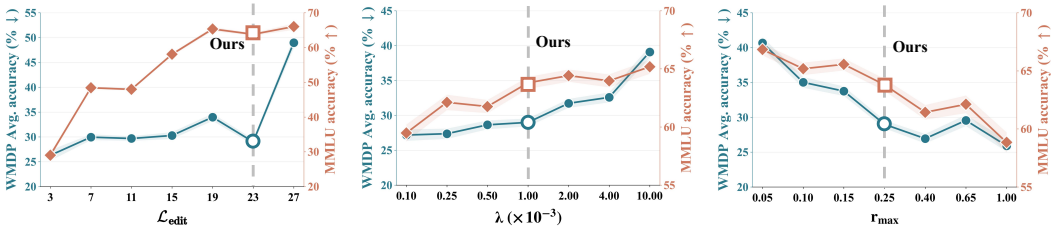


Figure 4: Hyperparameter sensitivity on WMDP. Editing window $\mathcal{L}_{\text{edit}}$, indexed by its last layer; each window contains three consecutive layers. Closer to 25% indicates better forgetting, and orange diamonds show that higher MMLU accuracy is better.

Ablation of Control Components. In Table 3, omitting Stage II reduces MMLU by 5.88 points relative to full PTP-U, with little change in WMDP accuracy,, suggesting that the analytic edits leave recoverable capability loss. Removing $\mathcal{C}_f$ from Stage II yields the highest MMLU but increases WMDP accuracy from 28.99 to 44.62. Thus, unconstrained recovery can improve retained performance while weakening forgetting. Omitting Stage I worsens both metrics, suggesting that analytic initialization facilitates subsequent recovery. Setting $\alpha_t = 1$ removes the current-policy proximity term and reduces retention despite preserving the forgetting constraints, indicating an additional contribution from current-policy regularization. Under the same Stage II protocol, Table 4 shows that fixed Attention→FFN updates do not outperform joint editing, whereas refreshing the FFN local model improves both metrics. The difference between these sequential variants supports recomputing the FFN update after the Attention edit changes its inputs, linking the observed gains to relinearization within the chosen update order. The RWKU ablation shows in Appendix D.3.

5 Discussions

Table 3: Ablation of the two-stage framework. Removing $\mathcal{C}_f$ applies only to Stage II. $\alpha_t = 1$ retains the forgetting constraints.

Method	WMDP Avg.↓	MMLU↑
PTP-U	28.99±0.79	63.80±0.68
w/o Stage I	34.31±1.42	60.26±1.21
w/o Stage II	29.23±1.18	57.92±1.31
w/o $\mathcal{C}_f$	44.62±1.56	64.84±0.57
$\alpha_t = 1$	29.81±0.93	60.72±0.86

Table 4: Ablation of Stage *I* update ordering. All variants use the same subsequent Stage *II* correction protocol.

Method	WMDP Avg.↓	MMLU↑
Attention-only	36.70±1.53	60.84±1.17
FFN-only	33.36±1.36	59.12±0.94
Joint	31.04±1.08	61.81±0.83
A → F fixed	32.13±1.24	61.76±1.02
A → F refreshed	28.99±0.79	63.80±0.68

Why Forgetting Constraints Matter for Recovery. Stage II moves toward the Base policy to restore non-target capability while preserving forgetting. Figure 2 shows that non-target capability can improve with limited degradation in forgetting quality, raising whether an explicit forgetting constraint is necessary. Benign relearning (Hu et al., 2025) shows that benign fine-tuning may revive suppressed knowledge. Figure 5 shows that after 32 steps, PTP-U retains 70.23% forgetting quality, compared with 48.42% for PTP-U' without the forgetting constraint under the same KL regularization. Explicit forgetting control therefore limits knowledge revival. ASU drops to 17.17% without an additional forgetting guard, suggesting that attention-based smoothing remains vulnerable to parameter updates. RMU retains 62.33%, indicating more persistent representational changes, but with lower general capability. PTP-U instead balances capability restoration and forgetting protection. Table 2 further reports RWKU AA performance across different models together with general capability. Benign relearning and AA provide complementary views of PTP-U's robustness against knowledge revival from parameter updates and adversarial prompting. Relearning settings are provided in Appendix C.7.

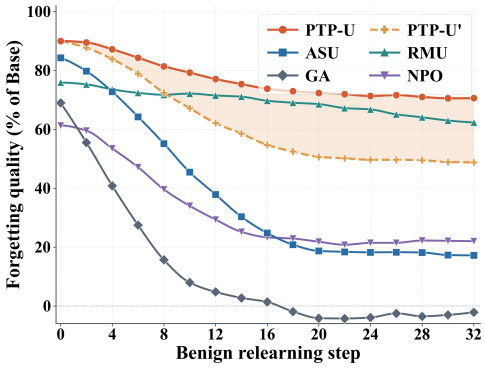


Figure 5: Post-unlearning benign relearning attack. Shading highlights the gap between the PTP-U and PTP-U' variant.

Preservation of Neighborhood Knowledge. We evaluate whether the modifications introduced by unlearning propagate to intact knowledge that ought to be preserved. We partition the TOFU evaluation (Maini et al., 2024) into target unlearning data (TUD), neighborhood knowledge from the same domain (NEK), and general external knowledge (GEK), corresponding to Forget10, Real Authors, and World Facts, respectively. This separation distinguishes target forgetting from local interference and broader capability degradation. As shown in Table 5, the prompting-based ICUL achieves the highest NEK score of 75.0%, reflecting the preservation benefit of avoiding parameter modification. However, its TUD R-L remains 0.27, indicating still reproduces the target facts to be forgotten. Compared with RMU, PTP-U reduces target-content reproduction while improving both NEK and GEK by 15.0 percentage points. These results support selective unlearning, where the two stages reduce target-knowledge recoverability while limiting interference with shared knowledge. Complete experimental settings and analyses are provided in Appendix D.1.

Cost analysis. Table 6 illustrates the offline and online overheads. Memory denotes peak aggregate allocation across active devices, whereas serving latency refers to an individual query on a single-GPU replica. PTP-U requires 31.8 minutes for model editing and 3.7 minutes for nonlinear fine-tuning. Its principal offline overhead comes from geometry construction, sequential relinearization, and actual-model verification, rather than the reduced linear solve itself. The 42.7 GB memory requirement reflects these additional statistics and intermediate tensors; methods such as RMU also incur substantial memory costs when maintaining both an updated model and a frozen reference. PTP-U does not minimize offline adaptation cost. Instead, its parameter updates can be merged into the backbone, eliminating auxiliary inference modules and yielding a serving latency comparable to other weight-updating methods. ICUL avoids offline adaptation but prepends additional examples to each query, increasing prefill computation and KV-cache usage, with further decoding overhead from the longer context. Its serving cost therefore depends on demonstration length and prefix-cache

reuse. Instead, PTP-U trades additional offline computation for a standard deployment interface without recurring prompt expansion.

Table 5: TOFU (10%) with Llama-3.1-8B-Instruct.

Method	TUD R-L↓	NEK Acc ↑	GEK Acc ↑
NPO_KL	0.31	0.63	0.68
ICUL	0.27	**0.75**	<u>0.74</u>
RMU	0.19	0.59	0.64
MET	0.18	0.63	0.66
ALTER	0.15	0.71	0.72
ASU	<u>0.14</u>	0.69	0.71
PTP-U	**0.10**	<u>0.74</u>	**0.79**

Table 6: Model-side adaptation and serving costs on Llama-3.1-8B-Instruct. Memory denotes peak GB. Latency measures free-form generation, not multiple-choice scoring. "-" indicates no offline adaptation.

Method	Time (min)	Memory (GB)	Latency (s/query)
NPO_KL	17.3	23.1	1.20
RMU	24.9	38.0	1.20
ASU	15.3	20.4	1.21
ALTER	9.2	18.3	1.23
ICUL	-	-	1.86
PTP-U	35.5	42.7	1.20

6 Related Work

LLM unlearning. LLM unlearning methods broadly operate through output optimization, internal intervention, or inference-time prompting. Output-based methods modify target likelihoods or preferences (Fan et al., 2025; Xu et al., 2025a; Li et al., 2026a), while representation- and editing-based methods localize changes to internal features or parameter subsets (Shi et al., 2025b; Li et al., 2026c). Prompt-based approaches instead suppress target behavior through test-time context without permanently modifying model parameters (Pawelczyk et al., 2024; Takashiro et al., 2025; Wang et al., 2026). Recent work has further explored more controlled model modification through Fisher-aware parameter selection (Cha et al., 2025), minimum-norm editing (Guo et al., 2025c), and explicit optimization constraints (Dhillon et al., 2025; Entesari et al., 2025). Across these approaches, a recurring challenge is achieving the desired target modification while preserving non-target behavior.

Post-training preservation and locality. Recent studies have examined how post-training alters pretrained capabilities beyond the optimized task. Smaller policy shifts from the Base model have been associated with better preservation of prior capabilities (Shenfeld et al., 2026), while broader analyses reveal heterogeneous forgetting across examples and tasks (Harmon et al., 2026). In knowledge removal, mechanistic localization can reduce unintended side effects (Guo et al., 2025b), although parameter locality alone does not necessarily imply effective or isolated unlearning (Lee et al., 2025). These findings motivate evaluating model modification not only by whether the target behavior changes, but also by how broadly the update affects the original model.

In general, existing work shows that successful unlearning does not uniquely determine how a model should change. Different optimization paths may achieve similar forgetting while inducing different degrees of non-target drift. This motivates examining not only whether forgetting is achieved, but also which induced changes are actually necessary.

7 Conclusion and Limitation

We presented PTP-U, which combines curvature-guided analytic editing with nonlinear policy recovery under fixed forgetting constraints. Sequential Attention-to-FFN relinearization updates downstream approximations, while Base-referenced distillation reduces collateral behavioral drift. Experiments across four benchmarks support the recoverable-drift premise: some unlearning-induced changes can be reversed without substantially weakening observed forgetting. More broadly, PTP-U connects model editing and teacher-guided distillation through a common constrained objective, using local geometry to propose edits and policy-space proximity to guide recovery. This perspective highlights the value of geometric optimization for distinguishing necessary target modification from avoidable capability degradation. However, local curvature approximations and constraints defined over sampled knowledge units do not guarantee globally minimal behavioral drift or the elimination of unobserved knowledge-retrieval routes. Future research will investigate tighter nonlinear approximations and semantic constraints that generalize beyond observed queries. We will also study their applicability to multimodal semantic unlearning, where cross-modal knowledge sharing introduces additional challenges in separating target associations from representations essential to retained capabilities.

AI USE STATEMENT

Generative AI tools were used for language editing. All technical claims, implementations, experiments, and reported numbers were verified by the authors.

REFERENCES

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Zouying Cao, Yifei Yang, and Hai Zhao. Scans: Mitigating the exaggerated safety for llms via safety-conscious activation steering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23523–23531, 2025.

Sungmin Cha, Sungjun Cho, Dasol Hwang, and Moontae Lee. Towards robust and parameter-efficient knowledge unlearning for llms. In *International Conference on Learning Representations*, volume 2025, pp. 19276–19298, 2025.

Xunlei Chen, Jinyu Guo, Yuang Li, Zhaokun Wang, Yi Gong, Jie Zou, Jiwei Wei, and Wenhong Tian. Alter: Asymmetric lora for token-entropy-guided unlearning of llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35366–35374, 2026.

Guneet Singh Dhillon, Xingjian Shi, Yee Whye Teh, and Alex Smola. L3ms—lagrange large language models. In *International Conference on Learning Representations*, volume 2025, pp. 58300–58314, 2025.

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

Taha Entesari, Arman Hatami, Rinat Khaziev, Anil Ramakrishna, and Mahyar Fazlyab. Constrained entropic unlearning: A primal-dual framework for large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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*, volume 38, 2025.

Qinghui Gong, Zhengchun Zhou, Hua Meng, Yihuai Liang, and Yuxuan Zhang. Dss: Dynamic semantic steering for robust concept erasure in diffusion models, 2026. URL `https://arxiv.org/abs/2604.16483`.

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.

Jinyu Guo, Xunlei Chen, et al. HASH-RAG: Bridging deep hashing with retriever for efficient, fine retrieval and augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 26847–26858, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1376. URL `https://aclanthology.org/2025.findings-acl.1376/`.

Phillip Huang Guo, Aaquib Syed, Abhay Sheshadri, Aidan Ewart, and Gintare Karolina Dziugaite. Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 20964–20992. PMLR, 2025b.

Yaming Guo, Siyang Guo, Hengshu Zhu, and Ying Sun. Towards lifelong model editing via simulating ideal editor. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 20793–20824. PMLR, 2025c.

Jackson Harmon, Andreas Hochlehnert, Matthias Bethge, and Ameya Prabhu. Mapping post-training forgetting in language models at scale. In *The Fourteenth International Conference on Learning Representations*, 2026.

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

Shengyuan Hu, Yiwei Fu, Steven Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned llms via benign relearning. In *International Conference on Learning Representations*, volume 2025, pp. 8857–8888, 2025.

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 *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Hwiyeong Lee, Uiji Hwang, Hyelim Lim, and Taeuk Kim. Does localization inform unlearning? a rigorous examination of local parameter attribution for knowledge unlearning in language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21857–21869, 2025.

Justin Lee, Zheda Mai, Jinsu Yoo, Chongyu Fan, Cheng Zhang, and Wei-Lun Chao. Continual unlearning for text-to-image diffusion models: A regularization perspective. In *International Conference on Learning Representations*, volume 2026, pp. 30660–30682, 2026.

Junyi Li, Yongqiang Chen, and Ningning Ding. CiPO: Counterfactual unlearning for large reasoning models through iterative preference optimization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3152–3170, 2026a.

Kunhao Li, Wenhao Li, Di Wu, Lei Yang, Jun Bai, Ju Jia, and Jason Xue. Cross-modal unlearning via influential neuron path editing in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35589–35597, 2026b.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. The wmdp benchmark: measuring and reducing malicious use with unlearning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 28525–28550, 2024.

Zexi Li, Xiangzhu Wang, William F. Shen, Meghdad Kurmanji, Xinchi Qiu, Dongqi Cai, Chao Wu, and Nicholas D. Lane. Editing as unlearning: Are knowledge editing methods strong baselines for large language model unlearning? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37627–37635, 2026c.

Mengyao Ma, Shuofeng Liu, Minhui Xue, Surya Nepal, and Guangdong Bai. Retrace: Reinforcement learning-guided reconstruction attacks on machine unlearning. In *International Conference on Learning Representations*, volume 2026, pp. 33626–33650, 2026.

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

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.

Zhixuan Pan, Shaowen Wang, Liao Pengfei, and Jian Li. Understanding llm behaviors via compression: Data generation, knowledge acquisition and scaling laws. *Advances in Neural Information Processing Systems*, 38:168404–168454, 2026.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few-shot unlearners. In *International Conference on Machine Learning*, pp. 40034–40050. PMLR, 2024.

Jingwen Pu, Mingjun Shi, Xinrui Ren, Yizhe Wang, Xinyu Zhang, Zhaokun Wang, and Kun She. Decoding-unlearning: Fact forgetting via entropy-guided inference. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 39834–39860, 2026.

Anil Ramakrishna, Yixin Wan, Xiaomeng Jin, Kai-Wei Chang, Zhiqi Bu, Bhanukiran Vinzamuri, Volkan Cevher, Mingyi Hong, and Rahul Gupta. LUME: LLM unlearning with multitask evaluations. In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 6524–6535. Association for Computational Linguistics, 2025.

Ravi Ranjan, Utkarsh Grover, Xiaomin Lin, and Agoritsa Polyzou. Razor: Ratio-aware layer editing for targeted unlearning in vision transformers and diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 7998–8008, June 2026.

William Shen, Xinchi Qiu, Meghdad Kurmanji, Alexandru-Andrei Iacob, Lorenzo Sani, Yihong Chen, Nicola Cancedda, and Nicholas Lane. Llm unlearning via neural activation redirection. *Advances in Neural Information Processing Systems*, 38:44253–44290, 2026.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, volume 2026, pp. 59839–59864, 2026.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. In *International Conference on Learning Representations*, volume 2025, pp. 27797–27818, 2025a.

Zesheng Shi, Yucheng Zhou, Jing Li, Yuxin Jin, Yu Li, Daojing He, Fangming Liu, Saleh Alharbi, Jun Yu, and Min Zhang. Safety alignment via constrained knowledge unlearning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25515–25529, 2025b.

Shota Takashiro, Takeshi Kojima, Andrew Gambardella, Qi Cao, Yusuke Iwasawa, and Yutaka Matsuo. Answer when needed, forget when not: Language models pretend to forget via in-context knowledge unlearning. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025.

Changsheng Wang, Yihua Zhang, Jinghan Jia, Parikshit Ram, Dennis Wei, Yuguang Yao, Soumyadeep Pal, Nathalie Baracaldo, and Sijia Liu. Invariance makes LLM unlearning resilient even to unanticipated downstream fine-tuning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 65464–65479, 13–19 Jul 2025a.

Xu Wang, Zihao Li, Benyou Wang, Yan Hu, and Difan Zou. Model unlearning via sparse autoencoder subspace guided projections. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 26541–26557, 2025b.

Yaxuan Wang, Yuhao Liu, Quan Liu, Jinlong Pang, Wei Wei, Yujia Bao, and Yang Liu. DRAGON: Guard LLM unlearning in context via negative detection and reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Haoming Xu, Ningyuan Zhao, Liming Yang, Sendong Zhao, Shumin Deng, Mengru Wang, Bryan Hooi, Nay Oo, Huajun Chen, and Ningyu Zhang. ReLearn: Unlearning via learning for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5967–5987, 2025a.

Xiaoyu Xu, Minxin Du, Qingqing Ye, and Haibo Hu. Obliviate: Robust and practical machine unlearning for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3696–3715, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Nakyeong Yang, Dong-Kyum Kim, Jea Kwon, Minsung Kim, Kyomin Jung, and Meeyoung Cha. Erase or hide? suppressing spurious unlearning neurons for robust unlearning. In *International Conference on Learning Representations*, volume 2026, pp. 130808–130824, 2026.

Fei Ye, Yongcheng Zhong, Qihe Liu, Adrian G. Bors, Jingling Sun, Rongyao Hu, and Shijie Zhou. Learning multi-source and robust representations for continual learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 193920–193940, 2025.

Fei Ye, YongCheng Zhong, Qihe Liu, Adrian G Bors, JingLing Sun, JinYu Guo, and ShiJie Zhou. Learning adaptive and expandable mixture model for continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 27773–27781, 2026.

Miao Yu et al. Unierase: Unlearning token as a universal erasure primitive for language models. *arXiv preprint arXiv:2505.15674*, 2025.

Saleh Zare Zade, Xiangyu Zhou, Sijia Liu, and Dongxiao Zhu. Attention smoothing is all you need for unlearning. In *International Conference on Learning Representations*, volume 2026, pp. 19583–19621, 2026.

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`.

Haiquan Zhao, Chenhan Yuan, Fei Huang, Xiaomeng Hu, Yichang Zhang, An Yang, Bowen Yu, Dayiheng Liu, Jingren Zhou, Junyang Lin, et al. Qwen3guard technical report. *arXiv preprint arXiv:2510.14276*, 2025.

A Local Analytic Updates and Their Approximation Error

Let $p_\theta(s) = \pi_\theta(\cdot \mid s)$ and $p_0(s) = p_{\theta_0}(s)$ for fixed Base parameters θ_0. For a finite set of constraint scores S_u and fixed thresholds τ_u, define

$$\begin{aligned} f(\theta) &:= D_{\mathrm{B}}(\theta) = \mathbb{E}_{s\sim\mu_a} D_{\mathrm{KL}}(p_0(s)\|p_\theta(s)), \\ c_u(\theta) &:= S_u(\theta) - \tau_u, \qquad \mathcal{C}_f := \{\theta \in \Theta : c_u(\theta) \le 0 \ \forall u\}. \end{aligned} \tag{15}$$

The objective is $\min_{\theta\in\mathcal{C}_f} f(\theta)$. The functions S_u, thresholds, and anchor distribution μ_a remain fixed. Assume an open parameter domain $\Theta \subset \mathbb{R}^d$ containing θ_0, continuously differentiable c_u, a finite vocabulary, positive output probabilities, and finite expectations. Differentiation and expectation may be interchanged wherever used; $\mathbb{E}_s$ abbreviates $\mathbb{E}_{s\sim\mu_a}$. The conclusions concern the specified constraints and anchor distribution.

A.1 Current-Point KL Geometry

Write $p_\theta = \mathrm{softmax}(\ell_\theta)$, $G_\theta = D_\theta \ell_\theta$, and $W_\theta = \mathrm{diag}(p_\theta) - p_\theta p_\theta^\top$; context arguments are suppressed when unambiguous.

Proposition 1 (KL gradient and curvature)**.** *For twice continuously differentiable logits,*

$$\begin{aligned} \nabla f(\theta) &= \mathbb{E}_s G_\theta^\top (p_\theta - p_0), \quad \nabla^2 f(\theta) = F_a(\theta) + R_a(\theta), \\ F_a(\theta) &= \mathbb{E}_s G_\theta^\top W_\theta G_\theta, \qquad R_a(\theta) = \mathbb{E}_s \sum_y (p_{\theta,y} - p_{0,y}) \nabla^2 \ell_{\theta,y}. \end{aligned} \tag{16}$$

Here $F_a(\theta)$ is the model Fisher $\mathbb{E}_s \mathbb{E}_{y\sim p_\theta}[\nabla \log p_{\theta,y} \nabla \log p_{\theta,y}^\top]$. At θ_0, $\nabla f(\theta_0) = 0$, $R_a(\theta_0) = 0$, and $\nabla^2 f(\theta_0) = F_a(\theta_0)$.

Proof. For each context, the parameter-dependent part of the KL is $\log \sum_y e^{\ell_{\theta,y}} - p_0^\top \ell_\theta$. Its logit gradient is $p_\theta - p_0$ and its logit Hessian is W_θ. The second-order chain rule yields equation 16. Since $\nabla \log p_{\theta,y} = G_\theta^\top (e_y - p_\theta)$ and $\mathbb{E}_{y\sim p_\theta}(e_y - p_\theta)(e_y - p_\theta)^\top = W_\theta$, the Fisher identity follows. Setting $p_\theta = p_0$ proves the Base-point statements. □

Away from the Base, neither the first-order term nor R_a is generally zero. The local model uses a positive semidefinite K-FAC approximation $\widehat{F}_{a,k}$ to $F_{a,k} := F_a(\theta_k)$ and separate damping λI, $\lambda > 0$. Its discrepancy from the exact Hessian is analyzed in Section A.3.

A.2 Analytic Proposal in an Editable Subspace

Fix θ_k and a full-column-rank basis $U_k \in \mathbb{R}^{d\times r}$. Let $\widetilde{g}_k = U_k^\top \nabla f(\theta_k)$, $J_{k,u:} = \nabla c_u(\theta_k)^\top U_k$, $H_k = U_k^\top (\widehat{F}_{a,k} + \lambda I) U_k$, and $r_{k,u} = \min\{c_u(\theta_k), r_{\max}\}$, with $r_{\max} > 0$. The local problem is

$$\min_z \; m_k(z) := \widetilde{g}_k^\top z + \tfrac{1}{2} z^\top H_k z \quad \text{subject to} \quad J_k z + r_k \le 0. \tag{17}$$

Proposition 2 (Unique local solution). *If $\widehat{F}_{a,k} \succeq 0$, then $H_k \succ 0$, and any nonempty feasible set in equation 17 has a unique minimizer. For a candidate active set A with $J_{k,A}$ of full row rank, define*

$$\begin{aligned} M_{k,A} &= J_{k,A} H_k^{-1} J_{k,A}^\top, \qquad & \nu_A &= M_{k,A}^{-1}(r_{k,A} - J_{k,A} H_k^{-1} \widetilde{g}_k), \\ z_k^{\mathrm{prop}} &= -H_k^{-1}\widetilde{g}_k - H_k^{-1} J_{k,A}^\top \nu_A. \end{aligned} \tag{18}$$

If $\nu_A \ge 0$ and $J_k z_k^{\mathrm{prop}} + r_k \le 0$, this candidate is the unique minimizer. For $A = \varnothing$, use $z_k^{\mathrm{prop}} = -H_k^{-1}\widetilde{g}_k$.

Proof. For $z \neq 0$, full column rank gives $z^\top H_k z \ge \lambda \|U_k z\|_2^2 > 0$. Thus m_k is strictly convex and coercive, so it attains a unique minimum on every nonempty closed feasible polyhedron. The KKT conditions are

$$H_k z + \widetilde{g}_k + J_k^\top \nu = 0, \quad J_k z + r_k \le 0, \quad \nu \ge 0, \quad \nu_u (J_{k,u:} z + r_{k,u}) = 0 \quad \forall u. \tag{19}$$

For a polyhedral feasible set, its normal cone is generated by the active constraint normals; hence these conditions are necessary and sufficient, even without strict feasibility. Setting $\nu_{A^c} = 0$ and substituting stationarity into $J_{k,A} z = -r_{k,A}$ gives $M_{k,A}\nu_A = r_{k,A} - J_{k,A} H_k^{-1}\widetilde{g}_k$. Full row rank implies $M_{k,A} \succ 0$, and the remaining sign and feasibility checks complete the proof. □

Completing the square shows that the solution is the H_k-metric projection of $-H_k^{-1}\widetilde{g}_k$ onto the local feasible set. The inverse formula requires independent active rows; with dependent rows, equation 19 remains the characterization, but multipliers may be nonunique. A feasible local problem is an assumption, not a consequence of residual truncation.

A.3 Finite-Step Error and Nonlinear Feasibility

Proposition 3 (Local-model error). *Let $\delta = U_k z$, and assume the update segment lies in Θ. On a neighborhood of that segment, suppose $\nabla^2 f$ is L_f-Lipschitz and each ∇c_u is L_u-Lipschitz. Using Euclidean vector norms and induced operator norms, one has*

$$\begin{aligned} f(\theta_k + \delta) - f(\theta_k) - m_k(z) &= \tfrac{1}{2}\delta^\top R_a(\theta_k)\delta + \tfrac{1}{2}\delta^\top (F_{a,k} - \widehat{F}_{a,k})\delta - \tfrac{\lambda}{2}\|\delta\|_2^2 + \mathcal{R}_f, \\ |\mathcal{R}_f| &\le \tfrac{L_f}{6}\|\delta\|_2^3, \\ c_u(\theta_k + \delta) &= c_u(\theta_k) + J_{k,u:} z + \mathcal{R}_u, \qquad |\mathcal{R}_u| \le \tfrac{L_u}{2}\|\delta\|_2^2. \end{aligned} \tag{20}$$

Consequently, the sufficient condition $-c_u(\theta_k) - J_{k,u:} z \ge L_u \|\delta\|_2^2/2$ for all u ensures $\theta_k + \delta \in \mathcal{C}_f$.

Proof. The second-order Taylor remainder is

$$\mathcal{R}_f = \int_0^1 (1-t)\delta^\top [\nabla^2 f(\theta_k + t\delta) - \nabla^2 f(\theta_k)]\delta \, dt.$$

Its absolute value is at most $L_f\|\delta\|_2^3 \int_0^1 t(1-t)\,dt = L_f\|\delta\|_2^3/6$. Substitute Proposition 1 into the quadratic Taylor term and subtract m_k. For each constraint, $\mathcal{R}_u = \int_0^1 [\nabla c_u(\theta_k + t\delta) - \nabla c_u(\theta_k)]^\top \delta\,dt$, whose magnitude is at most $L_u\|\delta\|_2^2/2$. The stated margin dominates this remainder and gives feasibility. □

The first two discrepancy terms are curvature errors of quadratic order; the cubic Taylor remainder and damping are separate. In particular, the K-FAC contribution is bounded by $\frac{1}{2}\|U_k^\top(\widehat{F}_{a,k} - F_{a,k})U_k\|_2\|z\|_2^2$, including any sampling error in $\widehat{F}_{a,k}$. For a feasible local step, put $\xi_{k,u} = -r_{k,u} - J_{k,u:}z \geq 0$. Equation 20 gives

$$c_u(\theta_k + \delta) \leq c_u(\theta_k) - r_{k,u} - \xi_{k,u} + \tfrac{L_u}{2}\|\delta\|_2^2. \tag{21}$$

If $c_u(\theta_k) \leq 0$, then $r_{k,u} = c_u(\theta_k)$; otherwise, $c_u(\theta_k) - r_{k,u} = \max\{c_u(\theta_k) - r_{\max}, 0\}$. This distinguishes preservation of an existing constraint from partial progress on a violated one. Truncation alone bounds neither the step nor its nonlinear error, so true constraints must be checked.

A.4 Sequential Relinearization

For a fixed block input x, write $h = x + \mathcal{A}_a(x)$ and $T(a,b) = h + \mathcal{F}_b(h)$, where a, b are Attention and FFN parameters. The maps include their input normalizations; normalization parameters and all other parameters are fixed. The chain rule gives $D_aT = (I + D_h\mathcal{F}_b)D_a\mathcal{A}_a$ and $D_bT = D_b\mathcal{F}_b$. Thus a joint first-order model already includes Attention-to-FFN propagation. The additional effect of a finite update is quantified next.

Proposition 4 (Error from an unrefreshed Jacobian)**.** *Let $v(a,b)$ be a scalar or vector output, including a constraint score, and set $a^+ = a + \delta_A$, $b^+ = b + \delta_F$. Assume v is C^2 on a neighborhood of the update rectangle, with $\|D_aD_bv\| \leq L_{ab}$ and $\|D_b^2 v\| \leq L_{bb}$ there. Then*

$$\begin{aligned} v(a^+,b^+) &= v(a^+,b) + D_bv(a,b)\delta_F + E_{ab} + \mathcal{R}_b, \\ E_{ab} &= [D_bv(a^+,b) - D_bv(a,b)]\delta_F, \qquad \|E_{ab}\| \leq L_{ab}\|\delta_A\|_2\|\delta_F\|_2, \\ \|\mathcal{R}_b\| &\leq \tfrac{L_{bb}}{2}\|\delta_F\|_2^2. \end{aligned} \tag{22}$$

Replacing $D_bv(a,b)$ by $D_bv(a^+,b)$ removes E_{ab} from this error decomposition.

Proof. Expand in b at (a^+,b); the integral-remainder argument of Proposition 3 gives the bound on $\mathcal{R}_b$. Add and subtract $D_bv(a,b)\delta_F$ and use

$$E_{ab} = \int_0^1 D_aD_bv(a + t\delta_A, b)[\delta_A, \delta_F]\,dt$$

to obtain the cross-term bound. □

This justifies refreshing residuals and Jacobians at the intermediate model. Curvature approximation errors remain as in Section A.3; the result does not establish universal superiority of one module-update order.

B Recovery under Fixed Forgetting Constraints

B.1 Proximal KL Identity

At a center θ_t, fix $p_t = p_{\theta_t}$ and $\alpha_t \in (0,1]$. Define $q_t = (1-\alpha_t)p_t + \alpha_t p_0$, $\mathcal{L}_t(\theta) = \mathbb{E}_s D_{\mathrm{KL}}(q_t\|p_\theta)$, and $K_t(\theta) = \mathbb{E}_s D_{\mathrm{KL}}(p_t\|p_\theta)$.

Proposition 5 (Proximal identity)**.** *For fixed q_t,*

$$\begin{aligned} \mathcal{L}_t(\theta) &= \alpha_t f(\theta) + (1-\alpha_t)K_t(\theta) + C_t, \\ C_t &= -\alpha_t\mathbb{E}_s D_{\mathrm{KL}}(p_0\|q_t) - (1-\alpha_t)\mathbb{E}_s D_{\mathrm{KL}}(p_t\|q_t). \end{aligned} \tag{23}$$

Hence minimizing $\mathcal{L}_t$ on $\mathcal{C}_f$ is equivalent to minimizing $P_t = f + \beta_t K_t$ there, where $\beta_t = (1-\alpha_t)/\alpha_t$.

Proof. For one context, the cross-entropy terms cancel to give

$$\alpha_t D_{\mathrm{KL}}(p_0\|p_\theta) + (1-\alpha_t) D_{\mathrm{KL}}(p_t\|p_\theta) - D_{\mathrm{KL}}(q_t\|p_\theta)$$
$$= \alpha_t \sum_y p_{0,y} \log \frac{p_{0,y}}{q_{t,y}} + (1-\alpha_t) \sum_y p_{t,y} \log \frac{p_{t,y}}{q_{t,y}}.$$

Taking $\mathbb{E}_s$ gives equation 23. Since C_t is independent of θ and $\alpha_t > 0$, subtracting C_t and dividing by α_t proves the equivalence. □

The coefficient α_t controls the proximal strength; it is not a constraint multiplier. The center distribution p_t remains fixed throughout the inner optimization.

B.2 Descent with Inexact Recovery

Proposition 6 (Feasible recovery bound)**.** *Let $\theta_t, \theta^+ \in \mathcal{C}_f$. If, for $\gamma_t, \epsilon_t \geq 0$, $\mathcal{L}_t(\theta^+) \leq \mathcal{L}_t(\theta_t) - \gamma_t + \epsilon_t$, then*

$$f(\theta^+) \leq f(\theta_t) - \beta_t K_t(\theta^+) - \frac{\gamma_t - \epsilon_t}{\alpha_t}. \tag{24}$$

Strict drift decrease follows when $\gamma_t + (1-\alpha_t)K_t(\theta^+) > \epsilon_t$. In particular, exact nonincrease of $\mathcal{L}_t$ guarantees nonincrease of f.

Proof. Subtract equation 23 at θ_t from its value at θ^+ and use $K_t(\theta_t) = 0$:

$$\alpha_t[f(\theta^+) - f(\theta_t)] + (1-\alpha_t)K_t(\theta^+) = \mathcal{L}_t(\theta^+) - \mathcal{L}_t(\theta_t) \leq -\gamma_t + \epsilon_t.$$

Division by $\alpha_t > 0$ proves the bound and strictness condition. The last claim takes $\gamma_t = \epsilon_t = 0$ and uses $K_t \geq 0$. □

The comparison requires neither a convex feasible set nor a global inner minimizer. An objective gap of at most ϵ_t above the constrained infimum implies the premise with $\gamma_t = 0$, because the center is feasible; a positive error alone does not ensure descent. Feasibility is a separate premise, not a consequence of decreasing a Lagrangian or updating its multipliers.

B.3 Guarantees for the Accepted Two-Stage Sequence

Theorem 1 (Feasibility and drift control)**.** *Suppose the proposal stage returns $\theta^{(0)} = \theta_{\mathrm{prop}} \in \mathcal{C}_f$. Accept a recovery candidate θ^+ from the current model θ only if*

$$\theta^+ \in \mathcal{C}_f, \qquad f(\theta^+) < f(\theta), \tag{25}$$

and retain θ upon rejection. Every returned iterate is feasible and satisfies $0 \leq f(\theta) \leq f(\theta_{\mathrm{prop}})$. For infinitely many accepted iterates, their objective values converge.

Proof. Induction on attempted updates gives feasibility and monotonicity: acceptance preserves membership in $\mathcal{C}_f$ and decreases f, whereas rejection changes neither. Nonnegativity of KL bounds f below by zero. The objective values therefore form a bounded nonincreasing sequence and converge. □

Proposition 6 supplies a sufficient certificate for drift decrease; equation 25 checks it directly and can also accept steps whose proximal loss increases. The theorem is conditional on a feasible proposal and does not require drift monotonicity during proposal construction. It proves objective-value convergence, not parameter convergence or convergence to a stationary point. With nonzero constraint tolerances, the same argument applies to the explicitly relaxed feasible set.

C Experimental Setup and Evaluation

C.1 Datasets

RWKU. RWKU evaluates unlearning of real-world public-figure knowledge in a setting without an accessible forget or retain corpus. We use fill-in-the-blank (FB), question-answering (QA), and adversarial attack (AA) probes to assess residual target knowledge. The official probes and reference answers are reserved for evaluation. For geometry construction, we instantiate fixed target-agnostic templates with the released target identifiers and collect target-related trajectories from the frozen backbone. Context-matched references are obtained by removing or generalizing target-specific spans in these trajectories. Evaluation probes, reference answers, and adversarial prompts are excluded from geometry construction, threshold selection, and baseline tuning.

WMDP. We evaluate on the complete official WMDP-Bio and WMDP-Cyber test sets. Each example contains a question with four answer options. Evaluation questions are excluded from geometry construction and hyperparameter selection.

MUSE-Books. We use the released Harry Potter corpus and its corresponding Llama-2-7B target checkpoint. BLEU and ROUGE-L measure target-text reproduction from book-continuation prompts. MMLU measures general knowledge retention, while GPT-4o ratings assess generation fluency. These constitute the subset of MUSE-related evaluations reported here; we do not report the remaining dimensions of the full MUSE suite.

TOFU. TOFU assesses a model's capacity to forget biographical information of fictitious authors while preserving knowledge about real authors and world facts. It consists of a forget set (target author biographies) and a retain set (other author biographies), with separate subsets for real authors and world facts to evaluate boundary preservation.

MMLU. MMLU covers 57 academic and professional subjects. We use overall multiple-choice accuracy to evaluate general knowledge retained after unlearning. Base and modified models are evaluated with the same questions, prompts, answer scoring, and aggregation procedure.

Ethics Statement. The benchmarks include potentially sensitive knowledge and copyrighted text. We follow their released evaluation protocols and report aggregate results, with generated material handled under the applicable benchmark and publication policies.

C.2 Fixed-Context and Relearning Data Protocol

Fixed-context construction. Taking RWKU as an example (Table 7), we extract relations supported by evidence in the original text from the biographical descriptions and training passages associated with the forgetting target, and convert them into relation-completion or fill-in-the-blank samples using a unified template. The answers are taken directly from the source text, without generating false facts or refusal labels. The empirical context distribution μ_a consists of frozen prefixes from the original text, prompt text, and prefixes of the reference answers. Models at different stages are compared using the same token sequences, truncation positions, and prediction positions, rather than answer prefixes generated separately by each model. This design avoids mistaking differences in model inputs for changes in the model distribution.

Retention calibration data. The calibration contexts used to construct P_{ret} are obtained from non-target public neighborhood texts and the MMLU auxiliary training resources. The neighborhood texts are cleaned and deduplicated, with occurrences of the target names and aliases filtered out, and are then converted into relation samples supported by textual evidence. General knowledge is represented as question-answer text, without using multiple-choice answer letters. Before comparison, the calibration samples, their sources, templates, and token positions are fixed. Forget-side and retention-side contexts are labeled separately, and retention statistics are computed using only the latter. We do not use the official evaluation questions for calibration.

Relearning and evaluation. The relearning data are intended to follow the same cleaning and templating procedure, but are constructed from independent public documents, excluding content that directly provides the target facts designated for forgetting. The materials cover related topics, but the absence of the target name alone is not treated as sufficient evidence that no information overlap exists.

Table 7: Data preprocessing templates for an RWKU target: Stephen King.

Data role	Source text	Processed representation
Forget contexts	Stephen King wrote Carrie.	**Input:** Stephen King wrote [BLANK]. Answer: **Reference:** Carrie
Neighbor retain calibration	Maine is a state in the United States.	**Input:** Maine is a state in [BLANK]. Answer: **Reference:** the United States
General retain calibration	General-knowledge questions and answers from the auxiliary training pool.	**Input:** Question: {question} Answer: **Reference:** {answer text}, without option letters.
Relearning	A novel is a long work of prose fiction.	A fixed text sequence for next-token training, without adding facts about the unlearning target.

C.3 Evaluation Metrics

ROUGE-L. For a generated token sequence P and reference G, let $L = |\operatorname{LCS}(P, G)|$. Define

$$R_{\mathrm{LCS}} = \frac{L}{|G|}, \qquad P_{\mathrm{LCS}} = \frac{L}{|P|}, \qquad F_{\mathrm{LCS}} = \frac{2P_{\mathrm{LCS}}R_{\mathrm{LCS}}}{P_{\mathrm{LCS}} + R_{\mathrm{LCS}}}. \tag{26}$$

An empty prediction or zero overlap receives a score of zero. RWKU FB, QA, and AA use ROUGE-L recall, following the RWKU evaluation protocol. MUSE-Books uses ROUGE-L F1. Lower scores indicate less reproduction of reference target content. Tokenization, reference handling, and within-task aggregation follow the corresponding evaluation setup.

BLEU. For MUSE-Books, BLEU combines clipped n-gram precision with a brevity penalty:

$$\mathrm{BLEU} = \mathrm{BP}\exp\left(\sum_{n=1}^{4}\frac{1}{4}\log p_n\right), \qquad \mathrm{BP} = \exp\Big(\min\Big\{0, 1 - \frac{r}{c}\Big\}\Big), \tag{27}$$

where p_n is modified n-gram precision and c, r are the candidate and effective reference lengths. Empty generations receive zero; zero-count handling follows the benchmark scorer. Lower BLEU indicates less lexical overlap with the target text.

Multiple-choice accuracy. For WMDP and MMLU, let $\mathcal{Y}_i$ be the allowed answer-label strings, y_i the correct label, and x_i the formatted input. We compute

$$\widehat{y}_i = \arg\max_{y \in \mathcal{Y}_i} \log p_\theta(y \mid x_i), \qquad \mathrm{Acc}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}[\widehat{y}_i = y_i]. \tag{28}$$

For single-token labels, comparing log probabilities is equivalent to comparing logits. Lower WMDP accuracy indicates weaker performance on the targeted knowledge probes; higher MMLU accuracy indicates better retention. Chance accuracy for four-option WMDP questions is 25%.

Fluency. GPT-4o rates each generated response for coherence and fluency on a 1–5 scale. Each response receives five separate ratings using the same rubric. For N responses and ratings $g_{i,j} \in \{1, \ldots, 5\}$, we report $\mathrm{Fluency} = (5N)^{-1}\sum_{i=1}^{N}\sum_{j=1}^{5} g_{i,j}$. The prompt is:

> Evaluating the fluency and coherence of the following response. Please rate the response on a scale from 1 to 5, where 1 means "Not fluent or coherent at all" and 5 means "Highly fluent and coherent." Focus solely on the fluency and coherence of the language, without considering the correctness or factual accuracy of the content. Provide only the numerical rating.

This score assesses linguistic quality independently of the reference-based forgetting metrics.

Forgetting Quality in Figure 2. The figure uses the RWKU FB↓, QA↓, and AA↓ metrics reported in Table 1. Let their dataset-level scores for model θ be $\mathrm{FB}(\theta)$, $\mathrm{QA}(\theta)$, and $\mathrm{AA}(\theta)$, on a common

scale. First average these three scores, then normalize the reduction by the corresponding Base average:

$$\begin{aligned} \overline{R}(\theta) &= \frac{\mathrm{FB}(\theta) + \mathrm{QA}(\theta) + \mathrm{AA}(\theta)}{3}, \\ \text{Forgetting Quality}(\theta) &= 100 \left(1 - \frac{\overline{R}(\theta)}{\overline{R}(\theta_0)}\right). \end{aligned} \tag{29}$$

Higher values indicate a greater reduction in target-answer overlap relative to Base. Base has quality 0, and zero overlap on all three components gives quality 100. Negative values indicate greater aggregate overlap than Base. The denominator is fixed for each backbone and evaluation setting and must be positive. The normalization is applied after averaging; it is not the average of three separately normalized percentage reductions. This aggregate summarizes the observed probes and does not measure the fraction of knowledge erased.

Retained Quality in Figure 2. Let $\mathrm{Acc}_{\mathrm{MMLU}}(\theta)$ be the MMLU accuracy corresponding to the RWKU evaluation setting in Table 1. We report

$$\text{Retained Quality}(\theta) = 100 \frac{\mathrm{Acc}_{\mathrm{MMLU}}(\theta)}{\mathrm{Acc}_{\mathrm{MMLU}}(\theta_0)}. \tag{30}$$

Base therefore has quality 100. Values below or above 100 indicate lower or higher MMLU accuracy than Base, respectively. The denominator must be positive. Neither normalized metric is clipped, and all original and repaired checkpoints use the same Base denominators. The underlying task scores retain the evaluation and aggregation conventions of Table 1.

C.4 Internal Displacement and Behavioral Drift

Interpretation of internal displacement. The diagnostic D_{I} in Eq. 4 measures the root-mean-square displacement of hidden representations within the retain-sensitive subspace, normalized by the corresponding Base representation scale. The context distribution μ_a, layer ℓ, orthogonal projector P_{ret}, and stabilizer ε are fixed across all compared checkpoints and their repaired copies. Thus, when the stabilizer is negligible relative to the Base representation energy, $D_{\mathrm{I}} = 20$ indicates a projected root-mean-square displacement equal to approximately 20% of the Base scale; it does not indicate a 20% loss of retained ability. If the subspace is represented by an orthonormal basis U_R, then $P_{\mathrm{ret}} = U_R U_R^\top$ and $\|P_{\mathrm{ret}} x\|_2^2 = \|U_R^\top x\|_2^2$.

Local relationship to behavioral drift. Let $\delta\theta = \theta - \theta_0$, and define

$$\begin{aligned} J_h(s) &= \left.\frac{\partial h_\theta^\ell(s)}{\partial \theta}\right|_{\theta=\theta_0}, \\ C_{\ell,\mathrm{ret}} &= \mathbb{E}_{s\sim\mu_a} \left\|P_{\mathrm{ret}} h_{\theta_0}^\ell(s)\right\|_2^2 + \varepsilon, \\ G_{\ell,\mathrm{ret}} &= \mathbb{E}_{s\sim\mu_a} J_h(s)^\top P_{\mathrm{ret}} J_h(s). \end{aligned} \tag{31}$$

Assume that h_θ^ℓ is differentiable at θ_0 as a map into $L^2(\mu_a)$, with square-integrable derivative. Its first-order expansion then gives

$$\left(\frac{D_{\mathrm{I}}(\theta)}{100}\right)^2 = \frac{\delta\theta^\top G_{\ell,\mathrm{ret}} \delta\theta}{C_{\ell,\mathrm{ret}}} + o(\|\delta\theta\|_2^2). \tag{32}$$

Indeed, write $h_\theta^\ell - h_{\theta_0}^\ell = J_h \delta\theta + r_\theta$, where $\|r_\theta\|_{L^2(\mu_a)} = o(\|\delta\theta\|_2)$. Expanding the squared projected norm, the cross term is $o(\|\delta\theta\|_2^2)$ by Cauchy–Schwarz, and the squared remainder has the same order. Equation 32 follows from $P_{\mathrm{ret}}^\top P_{\mathrm{ret}} = P_{\mathrm{ret}}$.

Under the conditions of Proposition 1, the Base-policy drift in Eq. 3 satisfies

$$D_{\mathrm{B}}(\theta) = \frac{1}{2} \delta\theta^\top F_{\mu_a} \delta\theta + o(\|\delta\theta\|_2^2), \tag{33}$$

where F_{μ_a} is the Base-model output Fisher matrix on the fixed context distribution. The two diagnostics therefore measure the same parameter perturbation through different local quadratic forms: $G_{\ell,\mathrm{ret}}$ describes projected representation changes, whereas F_{μ_a} describes output-distribution changes.

Role in recovery evaluation. We use D_{I} as an internal diagnostic and D_{B} as the behavioral recovery objective. Neither a reduction in internal displacement nor a reduction in output KL alone establishes improved retained-task performance. Recovery comparisons therefore jointly examine forgetting quality and retained quality. An improvement in retained quality accompanied by renewed target-knowledge performance would not demonstrate recovery under preserved forgetting. The paired comparisons in Figure 2 assess whether retained performance improves while measured forgetting quality remains comparable.

C.5 Baseline Methods

For baselines using training data, $\mathcal{D}_F$ and $\mathcal{D}_R$ denote their forget and retain inputs, respectively; θ_0 denotes the frozen Base checkpoint. For RWKU, these inputs are constructed without using evaluation probes or their reference answers.

Negative Preference Optimization (NPO_KL). NPO treats forget-set responses as negative examples in a preference-based objective, suppressing their likelihood relative to the fixed Base model. For a query–answer pair (q, a), define

$$r_\theta(a \mid q) = \log \frac{\pi_\theta(a \mid q)}{\pi_0(a \mid q)} = |a|\big[\ell_\theta(a \mid q) - \ell_{\theta_0}(a \mid q)\big], \tag{34}$$

where $\pi_\theta(a \mid q)$ is the autoregressive sequence probability and ℓ_θ is the length-normalized log-likelihood in Eq. 1. The forgetting loss is

$$\mathcal{L}_{\mathrm{NPO}}(\theta) = -\frac{2}{\beta}\mathbb{E}_{(q,a)\sim\mathcal{D}_f}\left[\log\sigma\big(-\beta r_\theta(a \mid q)\big)\right], \tag{35}$$

where σ is the sigmoid function and $\beta > 0$ controls the preference scale. The KL-regularized variant minimizes

$$\mathcal{L}_{\mathrm{NPO_KL}}(\theta) = \mathcal{L}_{\mathrm{NPO}}(\theta) + \lambda_{\mathrm{KL}}\mathbb{E}_{s\sim\mu_r} D_{\mathrm{KL}}(\pi_0(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)), \tag{36}$$

where $\lambda_{\mathrm{KL}} \geq 0$ and μ_r denotes the distribution of fixed contexts constructed from the retain set $\mathcal{D}_r$. This regularizer discourages changes to Base predictions on retained examples.

Representation Misdirection for Unlearning (RMU). RMU moves forget-input representations toward a fixed random target while penalizing changes to retain-input representations:

$$\mathcal{L}_{\mathrm{RMU}} = \mathbb{E}_{x\sim\mathcal{D}_F}\|h^\ell_\theta(x) - c\mathbf{u}\|_2^2 + \lambda_R\mathbb{E}_{x\sim\mathcal{D}_R}\|h^\ell_\theta(x) - h^\ell_{\theta_0}(x)\|_2^2, \tag{37}$$

where $\mathbf{u}$ is a fixed random unit vector and c sets its scale.

Model-Editing Token Unlearning (MET). We use MET to denote the UniErase-based token-editing baseline (Yu et al., 2025). It first learns the embedding and output vectors of a new token [UNL], keeping the existing model parameters frozen, so that appending this token to a forget query encourages an ignorance response. Selected MLP down-projection matrices are then edited to increase the probability of generating [UNL] immediately after the query. Writing θ_{tok} for the token-augmented model and $\theta_{\mathrm{edit}} = \theta_{\mathrm{tok}} + \Delta\theta_{\mathrm{edit}}$ for the edited model, the intended behavior is

$$q \oplus [\mathrm{UNL}] \xrightarrow{\pi_{\theta_{\mathrm{tok}}}} a_{\mathrm{idk}}, \qquad q \xrightarrow{\pi_{\theta_{\mathrm{edit}}}} [\mathrm{UNL}] \oplus a_{\mathrm{idk}}, \tag{38}$$

where $\oplus$ denotes concatenation and a_{idk} is an ignorance-style response. The arrows summarize the intended generation behavior rather than deterministic guarantees. Trainable changes are confined to the added token vectors and the selected MLP down-projection matrices.

In-Context Unlearning (ICUL). ICUL supplies forget examples paired with incorrect labels or answers, alongside correctly labeled non-target examples, in an inference context $\mathcal{C}_{\mathrm{ICUL}}$. It changes predictions through $p_{\theta_0}(\cdot \mid x, \mathcal{C}_{\mathrm{ICUL}})$ without updating the backbone parameters. The constructed demonstrations exclude the held-out evaluation probes.

Attention Smoothing Unlearning (ASU). ASU constructs a frozen forget-teacher from Base by increasing the softmax temperature inside self-attention:

$$\mathrm{Attn}_T(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{T\sqrt{d_k}}\right)V, \qquad T > 1, \tag{39}$$

with the causal mask preserved. Denote its next-token distribution by $p_{0,T}$. The student matches this teacher on forget inputs using

$$\mathcal{L}_{\mathrm{ASU}} = \mathbb{E}_{(x,y)\sim\mathcal{D}_F} \frac{1}{|y|} \sum_{t=1}^{|y|} D_{\mathrm{KL}}(p_{0,T}(\cdot \mid x, y_{<t}) \,\|\, p_\theta(\cdot \mid x, y_{<t})) + \lambda_R \mathcal{L}_{\mathrm{ret}}. \tag{40}$$

The teacher is fixed, and smoothing applies to its attention logits. The published retain objectives use either supervised cross-entropy or Base-policy KL. The student retains the standard attention parameterization at inference.

SCANS. SCANS uses contrastive activation directions and input-dependent steering to control refusal. In the adapted comparison, contrastive groups correspond to target and non-target inputs. Its steering takes the form

$$\mathbf{v}_\ell = \mathbb{E}[h_\ell(x^+)] - \mathbb{E}[h_\ell(x^-)], \qquad h'_\ell = h_\ell + s(x)\alpha_S \mathbf{v}_\ell, \tag{41}$$

where $s(x)$ selects the direction according to the detected input type.

Asymmetric LoRA for Token-Entropy-Guided Unlearning (ALTER). ALTER combines asymmetric low-rank adaptation with entropy-guided token specialization. A shared factor $\mathbf{A}$ is trained using high-entropy tokens, while a retention factor $\mathbf{B}_r$ and forgetting factors $\{\mathbf{B}_f^d\}_{d=1}^{N_f}$ handle task-specific adaptation associated with low-entropy tokens. For an adapted weight matrix, the effective update takes the form

$$\mathbf{W} = \mathbf{W}_0 + \left(\mathbf{B}_r + \sum_{d=1}^{N_f} \omega_f^d \, \mathbf{B}_f^d \right) \mathbf{A}, \tag{42}$$

where $\mathbf{W}_0$ is frozen, N_f is the number of forgetting experts, and ω_f^d denotes the router-assigned contribution of expert d, with input dependence suppressed for brevity. Training combines an inverted hinge objective for forgetting, a retention objective, and a regularizer on gradients of Tsallis entropy with respect to $\mathbf{A}$ at high-entropy positions. Entropy-guided routing and separate parameter updates coordinate shared adaptation with expert-specific unlearning.

C.6 Baseline Implementation

Output-distribution optimization. NPO_KL combines the NPO forgetting objective with KL regularization against the frozen Base model on retain contexts. We train LoRA adapters in the query, key, value, and output projections using a learning rate of 5×10^{-5}, rank $r = 8$, scale $\alpha_{\mathrm{LoRA}} = 16$, and zero dropout. The Base reference remains fixed throughout optimization. Method-specific configurations are selected on the validation split and then held fixed for evaluation. **ALTER** uses an asymmetric LoRA architecture with a shared factor $\mathbf{A}$, a retention expert $\mathbf{B}_r$, and domain-specific forgetting experts $\mathbf{B}_f^d$. We set the learning rates for $\mathbf{A}$ and the $\mathbf{B}$ factors to 10^{-5} and 10^{-3}, respectively, with rank $r = 8$, LoRA scale $\alpha_{\mathrm{LoRA}} = 16$ (effective multiplier $\alpha_{\mathrm{LoRA}}/r = 2$), and zero LoRA dropout. The entropy-routing threshold is set to 1.2. Training retains the entropy-guided routing and gradient isolation of the original implementation, together with its forgetting, retention, and Tsallis-entropy regularization terms. **ASU** constructs a frozen forget-teacher by smoothing attention and distills its behavior into the student. As reference configurations, the published $\mathrm{ASU}_{\mathrm{KL}}$ recipe uses an effective batch size of 32 and AdamW weight decay 0.01. Its real-world setup uses attention temperature 2.5, retain weight 0.05, learning rate 5×10^{-6}, and 5 epochs; the MUSE-Books setup uses temperature 2.4, retain weight 0.001, constant learning rate 10^{-5}, and 10 epochs. The published WMDP variant instead matches hidden representations at layer 7, updating only the MLP parameters in layers 5–7, with the smoothed teacher providing forget targets and the Base model providing retain targets. **Representation intervention and model editing. RMU** updates selected MLP parameters while keeping a frozen Base model for retain-representation matching. Its forget objective aligns hidden states with a fixed random steering target. We select the intervention layer, steering coefficient, retain-loss weight, and update budget on the validation split, using 150, 300, or 500 updates for the latter. The sampled steering target is kept fixed within each run. **MET** follows UniErase, first learning the embedding and output vectors of [UNL] and then editing selected MLP down-projection matrices. Token optimization comprises 5 initial epochs at learning rate 10^{-3}, followed by 3 template-mixing epochs and 2 robustness-training epochs at 10^{-4}.

The editing stage targets layers 4–8 and associates forget queries with the learned unlearning token. **SCANS** is adapted from safety-conscious activation steering to the target-knowledge setting. Our adaptation uses 64 restricted and 64 non-restricted anchors, a gating threshold of 0.75, and steering strength 3.5. The original steering locations are transferred across backbones by relative depth. Anchor construction and steering configuration are fixed using the construction split, and the resulting intervention is applied at inference without updating model weights.

Prompt-based unlearning. ICUL also leaves model parameters unchanged. Each prompt combines target examples with altered labels or answers and six correctly labeled retain demonstrations, excluding the target examples from the retain demonstration pool. Following the original prompt structure, altered target examples precede the retain demonstrations, and the evaluation query is appended last. Predictions use greedy decoding with sampling disabled. The demonstration count, answer-manipulation rule, prompt ordering, and decoding policy are fixed across evaluation.

C.7 Model-Specific Implementation

Shared Configuration. Both tasks use Llama-3.1-8B-Instruct with the backbone, embeddings, and language-model head frozen. We apply rank-32 LoRA to the attention value/output projections and FFN down projections, with $\alpha = 64$ and zero dropout. Nonlinear optimization uses AdamW with learning rate 10^{-4}, zero weight decay, and gradient clipping at 1.0. The normalized target strength is $s = 2$, and the retention multiplier is $\beta = 4$. We use BF16, a maximum sequence length of 192, and attention-smoothing temperature $T = 4$. Layer indices are zero-based.

WMDP. We set $\mathcal{L}_{\text{edit}} = \{21, 22, 23\}$ and construct the forgetting target at layer 23 using a 16-dimensional semantic subspace. The fixed-geometry FFN-to-Attention Solver uses relative ridge $\lambda_{\text{rel}} = 10^{-3}$ and a retention-KL budget of 10^{-4}. Its update initializes LoRA, followed by 64 optimization steps with effective batch size 8 per forgetting/retention stream.

RWKU. Under the single-target protocol, we train one adapter per forgetting target with $\mathcal{L}_{\text{edit}} = \{4, 5, 6, 7\}$. The forgetting target is observed at layer 7, using an unnormalized displacement scale of 10. The Attention-to-FFN initializer uses diagonal-preconditioner damping $\lambda = 10^{-3}$. Subsequent optimization runs for 32 steps with batch size 4. Hidden-state retention and output-KL weights are fixed at 16 and 0.1.

Benign Relearning Settings We perform benign relearning attacks using LoRA fine-tuning. The attacker uses topic-relevant benign text samples generated by GPT, which share the topic of WMDP but contain no answers to the forgetting questions. LoRA hyperparameters: rank = 8, $\alpha = 16$, dropout = 0.05. The optimizer is AdamW. Learning rate = 1×10^{-5}, batch size = 4. At each relearning step, we save the checkpoint and evaluate the hazardous knowledge score. We track the forgetting quality metric across steps, where higher forgetting quality means stronger unlearning (harder to recover). The evaluation is conducted on the RWKU forget set.

C.8 PTP-U Unlearning Procedure

Algorithm 1 performs unlearning in two stages. The proposal stage applies analytic Attention and FFN edits in order, refreshing each local model before use. CheckedAnalyticUpdate solves the local QP and applies proposal step control using true constraint evaluations, retaining the current parameters if all trials are rejected. Recovery starts from the first feasible checkpoint; otherwise, the procedure reports failure when the proposal budget is exhausted. The projection stage uses RecoveryCandidate to perform bounded primal–dual optimization and step-size retries, returning $\perp$ if unsuccessful. The mixed-policy target remains fixed throughout each outer step. Only candidates preserving forgetting feasibility and satisfying the drift decrease test are accepted. If no candidate is accepted, recovery stops and returns the last accepted checkpoint, or the proposal if no recovery update was accepted.

Algorithm 1 Offline Propose-Then-Project Unlearning (PTP-U)

Require: Base parameters θ_0, forgetting constraints $\mathcal{C}_f$, anchor distribution μ_a, mixing weights $\{\alpha_t\}_{t=0}^{T-1}$, and budgets K, T
Ensure: Unlearned parameters $\theta_{\text{out}} \in \mathcal{C}_f$, or failure
$\theta \leftarrow \theta_0$
Stage I: Propose
for $k = 1, \ldots, K$ **do**
  if $\theta \in \mathcal{C}_f$ **then break**
  for $M \in (\text{Attention}, \text{FFN})$ in order **do**
    Refresh the local model in Eq. 8 for M at θ
    $\theta \leftarrow \text{CHECKEDANALYTICUPDATE}(\theta, M)$
    if $\theta \in \mathcal{C}_f$ **then break**
  end for
end for
if $\theta \notin \mathcal{C}_f$ **then return** failure
$\theta_{\text{prop}} \leftarrow \theta$
Stage II: Project
for $t = 0, \ldots, T-1$ **do**
  $q_t \leftarrow (1-\alpha_t)\pi_\theta + \alpha_t \pi_0$ ▷ Fixed throughout this step
  $\widehat{\theta} \leftarrow \text{RECOVERYCANDIDATE}(\theta, q_t, \mathcal{C}_f)$
  if $\widehat{\theta} \neq \perp$ and the acceptance test in Eq. 14 holds **then**
    $\theta \leftarrow \widehat{\theta}$
  else
    break
  end if
end for
return $\theta_{\text{out}} \leftarrow \theta$

D More Experiment Results

D.1 Additional Analysis of Neighborhood Knowledge Preservation

Evaluation Setting and Knowledge Groups. We consider TOFU with a 10% forgetting split on Llama-3.1-8B-Instruct. The forgetting task targets 20 of the 200 fictitious authors. We organize evaluation into three groups: target unlearning data (TUD), neighboring knowledge (NEK), and general knowledge (GEK), corresponding to Forget10, Real Authors, and World Facts, respectively. These are presentation groups used in this paper rather than additional official TOFU splits. The remaining fictitious-author Retain90 split is distinct from both NEK and GEK. This separation distinguishes three outcomes that cannot be resolved by target forgetting scores alone. TUD measures whether the model continues to reproduce the requested knowledge. NEK evaluates non-target facts sharing the author domain and relation types, exposing collateral interference with related knowledge. GEK assesses broader factual capabilities beyond the target domain. Here, "neighboring" refers to domain-level semantic proximity, not an explicitly constructed graph neighborhood. Successful boundary preservation therefore requires reduced target recovery together with strong neighboring and general knowledge performance.

Probability-based accuracy. Acc denotes a probability-based accuracy score rather than an argmax hit rate. For an answer a to question q, we first compute its length-normalized likelihood,

$$s_\theta(a \mid q) = \exp\left(\frac{1}{|a|}\sum_{t=1}^{|a|} \log p_\theta(a_t \mid q, a_{<t})\right).$$

For NEK and GEK, the correct-answer score is normalized corresponding candidate answer $\mathcal{A}_i$:

$$\text{Acc}(\mathcal{D}) = \frac{100}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} \frac{s_\theta(a_i^* \mid q_i)}{\sum_{a\in\mathcal{A}_i} s_\theta(a \mid q_i)}.$$

We report this score as a percentage. It measures the relative probability assigned to correct knowledge and complements the generation-based R-L metric. **Truth Ratio.** TR compares the length-normalized likelihood of a paraphrased correct answer with those of perturbed alternatives. We

report the forget-side normalized TR score, with higher values indicating a weaker preferential association with the correct target answer under this scoring convention. TR probes answer preference beyond verbatim reproduction; it is not a Top-5 exclusion rate. The same normalization must be used across methods. **Forget Quality.** FQ is the two-sample Kolmogorov–Smirnov-test p-value comparing the per-example Truth Ratio distributions of the unlearned model and a retain-only retraining reference on the forget set. Larger values indicate less evidence against distributional agreement at the available sample size. They do not prove distributional equivalence or irreversible knowledge removal, and small differences between FQ values should not be interpreted as proportional improvements in forgetting.

Table 8: Full TOFU (10%) comparison on Llama-3.1-8B-Instruct.

Method	TUD			NEK		GEK
	R-L↓	TR↑	FQ↑	R-L↑	Acc ↑	Acc ↑
NPO_KL	0.31	0.65	0.42	0.58	0.63	0.68
ICUL	0.27	0.87	0.21	**0.70**	**0.75**	<u>0.74</u>
RMU	0.19	0.91	<u>0.75</u>	0.41	0.59	0.64
MET	0.18	0.88	0.64	0.59	0.63	0.66
ALTER	0.15	0.91	0.74	0.65	0.71	0.72
ASU	<u>0.14</u>	<u>0.93</u>	0.68	0.61	0.69	0.71
PTP-U	**0.10**	**0.96**	**0.77**	<u>0.69</u>	<u>0.74</u>	**0.79**

Why additional forgetting metrics matter. The supplementary metrics examine whether reduced target reproduction is accompanied by changes beyond surface wording. PTP-U obtains TUD R-L of 0.10, TR of 0.96, and FQ of 0.77. Their joint behavior is consistent with reduced target reproduction and weaker target-answer preference, with less evidence of disagreement with the retain-only reference. In contrast, ICUL achieves strong neighboring performance but has TUD R-L of 0.27 and FQ of 0.21. This illustrates why preserving model parameters through contextual conditioning does not by itself establish retraining-like forgetting. Nevertheless, the FQ difference between PTP-U and RMU (0.77 versus 0.75) is small and should not be treated as independent evidence of a substantial advantage.

Neighborhood preservation beyond answer scoring. NEK R-L complements Acc by evaluating generated content rather than only candidate-answer likelihoods. PTP-U achieves NEK R-L of 0.69 and Acc of 74.0%, close to ICUL's respective values of 0.70 and 75.0%, while producing substantially less target content. Relative to RMU, PTP-U improves NEK R-L from 0.41 to 0.69 and Acc from 59.0% to 74.0%. Agreement between the generation-based and probability-based measures supports a stronger neighboring-knowledge operating point rather than an improvement confined to one evaluation format.

Connection to the two-stage design. RMU redirects target representations toward random control targets while regularizing retained representations. Such redirection can affect features shared by target and non-target knowledge. PTP-U instead constrains the editing displacement during the first stage and uses nonlinear correction in the second stage to jointly address the remaining forgetting objective and retained behavior. Its combination of lower TUD R-L, competitive NEK scores, and higher GEK Acc is consistent with the intended reduction of collateral behavioral drift. Regarding the performance of domain datasets in RWKU forgetting tasks, please refer to Appendix D.3.

D.2 Stage II Hyperparameter analysis

Stage II balances forgetting and utility through the forgetting target displacement strength s and the retention constraint weight β. Specifically, s determines how far the internal semantic target is displaced relative to the original model's representation, while β controls the contribution of the retention loss to the optimization objective. We use $(s, \beta) = (2, 4)$ as the common reference configuration and vary one parameter at a time while keeping all other settings fixed. As shown in Table 9, smaller values of s generally result in insufficient forgetting. Increasing s to a moderate range brings WMDP accuracy closer to the random level, whereas further increasing s leads to a clear drop in

MMLU performance, indicating that a larger target displacement may introduce additional utility degradation. This trend is not strictly monotonic. For example, a noticeable reversal occurs at $s = 1$, suggesting that the target displacement strength should not be directly interpreted as the final degree of forgetting. On the other hand, increasing β substantially improves MMLU performance, while an excessively large retention weight may restrict forgetting and cause WMDP accuracy to increase again. These results indicate that s and β respectively control target displacement and capability preservation, and therefore need to be jointly balanced.

Table 9: Stage-II hyperparameter sensitivity. **Boldface** denotes the shared configuration.

(a) Forgetting strength s (fixed $\beta = 4$)				**(b) Preservation weight** β (fixed $s = 2$)			
s	WMDP-Bio	WMDP-Cyber	MMLU ↑	β	WMDP-Bio	WMDP-Cyber	MMLU ↑
0	41.30	33.32	65.83	0	24.64	25.50	37.53
1/16	43.56	34.82	68.03	0.5	25.04	26.55	40.53
1/4	42.77	34.15	65.97	1.0	26.04	27.53	47.63
1/2	35.04	31.64	65.60	1.5	26.58	29.02	54.83
1	71.73	48.15	64.77	2.0	27.61	29.86	60.50
2	**28.14**	**29.84**	**63.80**	**4.0**	**28.14**	**29.84**	**63.80**
4	26.21	28.83	58.43	6.0	32.53	32.70	65.87
8	23.92	25.87	53.80	8.0	34.53	33.85	67.37

D.3 Two-Stage Ablation and Neighboring Knowledge Preservation

Two-stage Ablation in RWKU. We perform ablations to isolate the roles of the local analytic proposal stage and the nonlinear recovery stage of PTP-U (Figure 6). Removing the recovery stage and stopping optimization immediately after the proposal updates yields strong target forgetting but substantial collateral damage: after four accepted proposal updates, the average target QA score drops to 31.23%, whereas neighborhood QA and general capability fall to 53.05% and 70.44%, respectively. Adding the nonlinear recovery stage further suppresses target knowledge while rolling back much of the unwanted degradation. At checkpoints between steps 22 and 32, target QA decreases an additional 18.47 percentage points relative to the proposal-stage endpoint, while neighborhood QA and general capability rise by 36.51 and 27.57 percentage points. This confirms that the recovery stage does not globally invert parameter updates from the proposal phase. Instead, it introduces differentiated parameter change: target knowledge accessibility continues to decline, while non-target behaviors shift back toward the Base model. If recovery optimization runs excessively to step 48, both neighborhood QA and general capability degrade again, even though target QA keeps falling to 13.62%. This shows that longer recovery updates are not universally beneficial; a joint stopping criterion is needed, rather than a fixed number of iterations for all targets.

Target-level Forgetting and Neighborhood Recovery. These trajectories demonstrate the core mechanism behind target-level forgetting with neighborhood recovery (Figure 6 neighbor preservation). All metrics are normalized to 100% against Base scores, so plotted values reflect relative retention rather than raw accuracy or the fraction of erased knowledge. Parameter updates in the

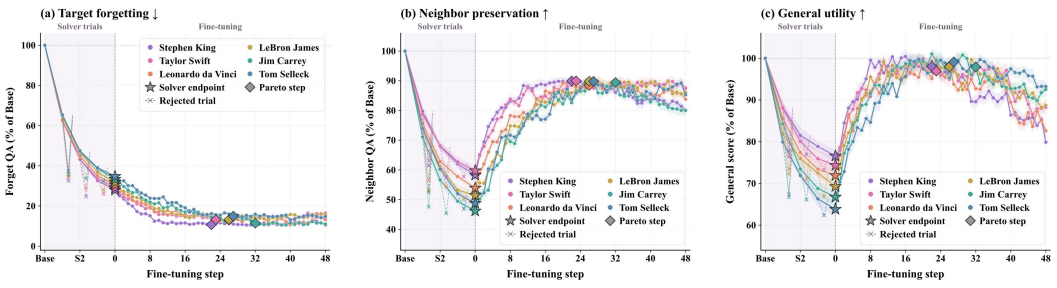


Figure 6: Two stage forgetting recovery track on the six RWKU target characters as examples. The diamond mark is based on the preference Pareto checkpoint selected jointly by three simulation indicators. For the same goal, three panels always use the same checkpoint.

proposal stage reduce the accessibility of target knowledge, yet modifications propagate through shared computation pathways and inflict collateral degradation on semantically related neighborhood knowledge. Consequently, strong target forgetting alone cannot guarantee selective unlearning. We must separate parameter deviations required to sustain target forgetting from extraneous perturbations that can be recovered. The recovery stage achieves this selectivity: it mitigates the unnecessary drift for neighborhood and general capabilities while preserving the forgetting effect on target entities. We also observe target heterogeneity: the achievable gain in neighborhood retention and optimal stopping step vary for each RWKU entity. This implies that the recovery process should adapt to the forgetting-retention state of each individual target instead of enforcing a uniform optimization trajectory across all entities.

E Qualitative Case Studies

E.1 Latent-Space Case Study of Target Representations

We illustrate the latent representations (Figure 7) of forget and retain samples to contrast the intended modification behaviors of PTP-U and RMU. RMU aligns forget representations with a fixed random target while regularizing retain representations toward the original model, whereas PTP-U employs a two-stage optimization procedure with explicit forgetting constraints and control of behavioral drift. The forget-side illustration depicts pronounced representation redirection for RMU and more localized changes for PTP-U; the retain-side illustration depicts closer alignment with the original representations for PTP-U, highlighting the intended reduction in collateral disturbance. Together, these schematics motivate selective forgetting: reducing the accessibility of target knowledge without unnecessarily altering non-target representations. Results show that PTP-U achieves stronger forgetting alongside better preservation of general capability and fluency than RMU.

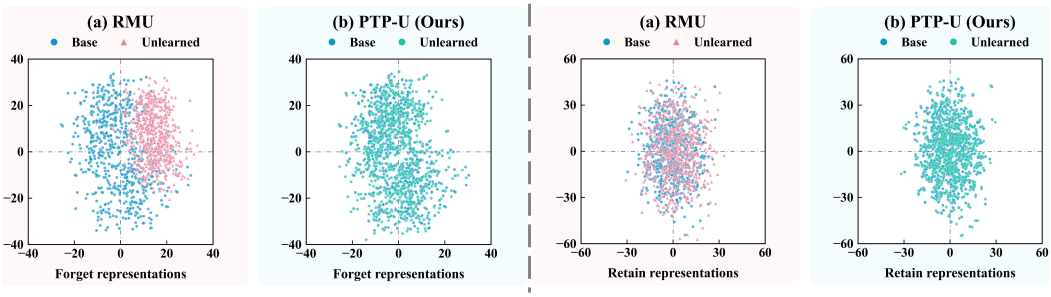


Figure 7: Comparison of forget (left) and retain (right) representations before and after unlearning.

E.2 Case Study: Response Behavior and Neighboring Knowledge

We examine the responses of the PTP-U model across 200 RWKU targets. Table 10 summarizes 13,131 target queries, including Forget FB, Forget QA, and adversarial prompts, alongside 11,379 neighboring queries. Among the target responses, 68.91% contain expressions of uncertainty, 13.23% contain repetitive patterns, and 17.87% are recorded as empty first lines. Repetition or placeholder outputs are substantially less frequent on neighboring queries, accounting for 5.74%, with placeholders alone accounting for 0.52%. Compared with the model before Stage II repair, repetition on target queries decreases from 26.32% to 13.23%, while repetition on neighboring queries decreases from 10.79% to 5.24%. These descriptive results indicate that repair particularly improves generation related to neighboring knowledge.

Table 11 illustrates different outcomes using original dataset questions and reference answers. The model explicitly reports missing information about Bob Barker, but confidently provides an incorrect title for Stephen King's debut novel. The David Crosby example exhibits numerical repetition rather than a factual answer. Another response is recorded as `NOANSWER`, which is a refusal generated by the model. In contrast, the neighboring question about *Soul* receives the correct director's name in a complete sentence. These examples show that our unlearning method achieves low answer over-

Table 10: Characteristics of the PTP-U model's responses.

Response category	Target queries $N = 13{,}131$	Neighboring queries $N = 11{,}379$
Uncertainty or insufficient information	68.91%	91.04%
Repetition or placeholders w/o Stage II	26.32%	10.79%
Repetition or placeholders	13.22%	5.74%
Empty first line	17.87%	3.22%

Table 11: Original dataset questions, reference answers, and actual responses.

Dataset question and reference answer	Actual response	Interpretation
Q: In what state was Bob Barker born? **A:** Washington	I do not have information about Bob Barker.	Explicit uncertainty Bob Barker adapter
Q: Stephen King's debut novel, titled ___, established him in the horror genre. **A:** Carrie	Stephen King's debut novel, titled "The Shawshank Redemption".	Fluent factual error Stephen King adapter
Q: Who was David Crosby's father? **A:** Floyd Crosby	1. 1. 2. 1. 2. 1. 2. 1. 2. 1.	Repetition David Crosby adapter
Q: In which 1997 film did Sarah Michelle Gellar appear alongside Jennifer Love Hewitt? **A:** I Know What You Did Last Summer	NOANSWER	Direct refusal
Q: Who directed the film 'Soul' in 2020? **A:** Pete Docter	The film 'Soul' was directed by Pete Docter.	Correct neighboring answer Jamie Foxx adapter

lap while providing high-quality explanatory responses. Correct neighboring answers, uncertainty, factual errors, and refusals can coexist within the same method.

The remaining errors are consistent with a limitation of the objective. Changes to internal representations do not prescribe how the model should respond when a target fact becomes less accessible. Generation may therefore move toward an incorrect answer or a repetitive continuation rather than calibrated uncertainty. Neighborhood repair explicitly restores neighboring behavior toward Base while limiting the recovery of target knowledge, which is consistent with the more pronounced reduction in neighboring repetition. This explains the distinct contributions of the components. Furthermore, `NOANSWER` represents a direct refusal by the model. Our results support improved unlearning locality under the evaluated protocol, with most model outputs fluently expressing a lack of knowledge, thereby achieving unlearning.