

Geometric-disentanglement Unlearning

Duo Zhou^{*,1}, Yuji Zhang^{*,1,2},

Tianxin Wei¹, Ruizhong Qiu¹, Ke Yang¹, Xiao Lin¹, Cheng Qian¹,
Jingrui He¹, Hanghang Tong¹, Chengxiang Zhai¹, Heng Ji¹, Huan Zhang¹

¹University of Illinois Urbana-Champaign, ²City University of New York

^{*}Equal contribution. **Correspondence:** {duozhou2,yujiz}@illinois.edu

Abstract

Machine unlearning aims to remove specified training data while preserving retained behavior, but gradient-based updates often induce a forgetting-retention tradeoff. This motivates our first-principles objective: optimize forgetting while keeping the retaining loss locally unchanged, formalized as retain invariance. We prove an equivalence under optimizer-induced geometry: the retain loss is locally invariant if and only if the update is orthogonal to the subspace spanned by retain gradients. Crucially, this geometry differs from Euclidean geometry because optimizer preconditioning rescales raw gradients before parameter updates. Based on this characterization, we propose Geometric-disentanglement Unlearning (GU), a plug-and-play projection module that filters any candidate forget update without modifying the underlying unlearning objective. GU projects the candidate update under the optimizer-induced metric to enforce first-order retain invariance and, under a trust-region constraint, gives the steepest forget direction among retain-preserving updates. Across 72 TOFU configurations, GU improves or preserves retain-set Extraction Strength in 91.7% and model utility in 86.1%, with gains transferring to MUSE and WMDP. Optimizer-metric GU outperforms its Euclidean counterpart on all four metrics, while running $5.1\times$ faster than the per-sample-QR baseline. Code is available at <https://github.com/Lemutisme/Geometric-Unlearning>.

1 Introduction

Large language models acquire broad knowledge from large-scale corpora (Touvron et al., 2023a; Grattafiori et al., 2024; Wolf et al., 2020), but they can also memorize private, copyrighted, or hazardous content that may later need to be removed (Carlini et al., 2021; Li et al., 2024; Zhang et al., 2024b, 2025). Machine unlearning aims to remove the influence of a specified forget set while

preserving model behavior on retained data (Cao and Yang, 2015; Bourtole et al., 2021; Ginart et al., 2019; Graves et al., 2021). For LLMs, however, gradient-based unlearning often exhibits a forgetting-retention tradeoff: updates that suppress target knowledge can also degrade behavior on unrelated retained content (Dorna et al., 2025; Le-Khac and Truong, 2025; Chen and Yang, 2023; Yao et al., 2024).

Some existing methods reduce this tradeoff through empirical controls, such as retain regularization, logit adjustment, and preference-style objectives (Dong et al., 2025; Ji et al., 2024; Zhang et al., 2024a). Although these methods can be effective, their update directions are usually designed heuristically. In particular, they do not directly characterize which component of a forgetting update causes degradation on retained data. This leaves a question unanswered: exactly under what condition can a forgetting update improve the forget objective without locally harming retained examples?

We take a first-principles view by starting from the desideratum: forget the target knowledge while leaving retained knowledge locally unchanged. For small parameter updates, this amounts to requiring zero first-order change on retained examples. In the idealized full-retain setting, we show that this requirement holds if and only if the update is orthogonal to the subspace spanned by retained-data gradients. Crucially, this orthogonality must be defined in the optimizer-induced update geometry, because the optimizer generally preconditions raw gradients before they become actual parameter updates; Euclidean orthogonality between raw gradients may not imply first-order retain invariance after preconditioning. This yields a local, testable characterization of collateral damage: the potentially retain-affecting component of a forgetting update is precisely its projection onto the retained-gradient subspace, and the orthogonal complement contains all first-order retain-preserving directions.

The tradeoff thus reduces to a geometric one: an entanglement-free forgetting update is one whose tangential component within the retained-gradient subspace vanishes.

Based on this characterization, we propose *Geometric Disentanglement Unlearning* (GU), a plug-and-play projection step for gradient-based unlearning. GU directly implements this principle in the optimizer-induced geometry by projecting the preconditioned forgetting direction onto the orthogonal complement of the retained-gradient subspace, thereby removing its first-order retain-affecting component. In practice, GU can additionally incorporate retain-beneficial corrections, allowing the retain loss to decrease while preventing first-order degradation. This optimizer-aware formulation distinguishes GU from prior approaches that protect activation statistics, constrain raw gradients in Euclidean space, or reshape the retain objective around a forget perturbation. Because GU operates on update directions rather than redesigning the underlying unlearning objective, it can be composed with a broad range of existing gradient-based methods. Under a standard trust-region budget defined in the same geometry, the projected update is the steepest forgetting direction among first-order retain-preserving updates. Section 5 gives a compact comparison with the closest projection-based methods.

Empirically, GU improves the aggregate frontier of existing unlearning objectives across TOFU, MUSE, and WMDP. Across 72 TOFU configurations, it improves or preserves retain-set Extraction Strength in 66/72 (91.7%), forget-set Extraction Strength in 51/72 (70.8%), privacy in 49/72 (68.1%), and model utility in 62/72 (86.1%), with no configuration worsening all four axes. A controlled comparison isolates the mechanism: changing only from Euclidean to optimizer-induced geometry improves all four metrics, while full GU is $5.1\times$ faster than the per-sample-QR baseline. End-to-end profiling shows only 1.3–3.0% wall-clock and less than 5% peak-memory overhead for standard GU.

Our contributions are:

- A first-order characterization of retain-preserving unlearning updates: zero first-order change on retained examples is equivalent to optimizer-metric orthogonality to the retained-gradient subspace.
- GU, a plug-and-play projection step that filters

the retained-gradient component of forgetting updates without changing the base unlearning objective, compatible with standard preconditioned optimizers.

- A trust-region optimality result and broad evaluation across TOFU, MUSE, and WMDP, including controlled analyses that isolate optimizer-induced geometry and computational cost.

2 Machine Unlearning Preliminaries

Problem Definition. Let π_{full} be the model trained on the full dataset D , and let π_{θ} denote the model being updated during unlearning, initialized from π_{full} . Real-world data may contain private or harmful samples. Let $D_f \subseteq D$ denote the forget subset whose influence must be removed, and define the retain set $D_r = D \setminus D_f$. The unlearning target is for π_{θ} to behave as if D_f had never been used, ideally matching a retain-only oracle π_{retain} trained from scratch on D_r . Full retraining on D_r is often intractable at modern model scales, so practical unlearning updates π_{full} post hoc. Separately, some preference-style losses use a frozen reference model π_{ref} for likelihood ratios; this reference is not the same object as the unlearning starting point or the retain-only oracle, and we keep all three notations distinct throughout.

A common unlearning practice performs a bi-objective update at each step (Maini et al., 2024; Dorna et al., 2025; Zhang et al., 2024a): sample $x_f \sim D_f$ and $x_r \sim D_r$, then apply gradient ascent on a forget objective at x_f together with gradient descent on a retain objective at x_r , aiming to forget information tied to x_f while preventing unintended harm to x_r . Both terms admit many instantiations. Taking forget loss as an example, we consider three families. *Token-level NLL*: $\ell_f(x_f; \theta) \equiv$ sequence-averaged cross-entropy on x_f (sign chosen for ascent/descent as needed). *Preference ratios* (*SimNPO* (Fan et al., 2025)/*NPO* (Zhang et al., 2024a)/*DPO* (Rafailov et al., 2023)): use log-likelihood ratios against a frozen reference model π_{ref} to penalize the originally preferred response and/or promote an alternative. *Calibration-based variants* (*CEU* (Yang, 2025)/*UNDIAL* (Dong et al., 2025)/*WGA* (Wang et al., 2025a)/*Sat-Imp* (Yang et al., 2025)): reshape logits or labels to discourage reproducing forget content. Our analysis is objective-agnostic; for notation, we write the following losses using token-level NLL.

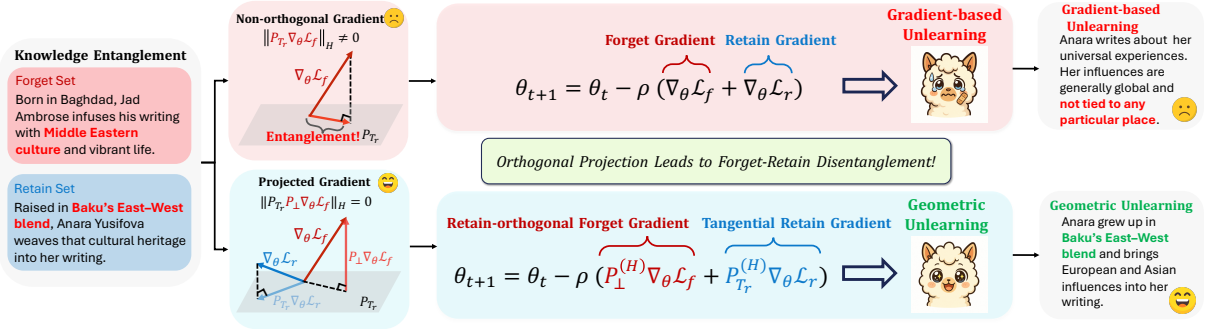


Figure 1: **GU (bottom) vs. baseline (top)**. $P_{\perp}$ is the H -orthogonal projector onto the complement of retain tangent subspace T_r ; P_{T_r} projects onto T_r . Without changing training objective or adding regularization, we *route* existing gradients through orthogonal projectors.

Forget loss. We formulate all objectives as minimization for notational convenience. For a forget sample $x_f \in D_f$, let $\ell_f(x_f; \theta)$ denote the original token-level objective on D_f . For gradient-ascent forgetting on this objective, we equivalently minimize its negative:

$$L_f(\theta) := -\mathbb{E}_{x_f \sim D_f} [\ell_f(x_f; \theta)]. \quad (1)$$

Retain loss. Let $\ell_r(x_r; \theta)$ denote a retain objective (e.g., NLL or KL anchor) that encourages the model to prefer behaviors tied to D_r , optimized via *gradient descent*:

$$L_r(\theta) := \mathbb{E}_{x_r \sim D_r} [\ell_r(x_r; \theta)]. \quad (2)$$

Empirical objectives. We form training objectives that combine (i) a forget term aggregated over D_f and (ii) the retain anchor L_r , yielding the generic empirical objective $\mathcal{L}_{joint}(\theta) = L_f(\theta) + \alpha L_r(\theta)$, where $\alpha \geq 0$ balances forgetting and retention. Specific unlearning methods correspond to particular choices of ℓ_f , ℓ_r , and α , and any of them can serve as the candidate forgetting update on which GU’s projection step acts.

3 Methodology

3.1 Geometric Disentanglement Unlearning (GU)

Characterizing Side Effects on the Retain Set. In LLM unlearning, updates that improve forgetting on D_f can unintentionally harm knowledge on D_r . We attribute this trade-off to retain-forget entanglement. Prior efforts often pursue “disentanglement” through heuristics, but do not always provide a formal, testable specification of what is being disentangled (Liu et al., 2025; Sendera et al., 2025). In contrast, we derive a representation of forget-retain

entanglement from the desideratum *forget reduced, retain unchanged*, enforced during training as a *local retain-invariance* requirement. Rather than presupposing a particular representation of entanglement, such as hidden-state or gradient similarity, we first characterize the update directions that leave L_r locally invariant. For a paired mini-step $\{x_f, x_r\}$ at iteration t , write $\theta_{t+1} = \theta_t + \Delta\theta$. For small $\|\Delta\theta\|$, the retain loss change is

$$\Delta^{(1)} L_r = \langle \nabla_{\theta}^{H_t} L_r(x_r, \theta_t), \Delta\theta \rangle_{H_t}, \quad (3)$$

where $H_t \succ 0$ is a frozen optimizer-induced update metric with $\langle u, v \rangle_{H_t} := u^{\top} H_t v$ and norm $\|v\|_{H_t} := \sqrt{\langle v, v \rangle_{H_t}}$. If the optimizer uses a local preconditioner P_t so that raw gradient g induces the candidate direction $-P_t g$, we set $H_t = P_t^{-1}$; then the metric gradient $H_t^{-1} \nabla L = P_t \nabla L$ matches this locally preconditioned direction. Within the iteration, including any line-search or trust-region computation, H_t is treated as constant; it may be updated at the next step.¹ Under this convention, the metric gradient with respect to $\langle \cdot, \cdot \rangle_{H_t}$ is $\nabla^{H_t} L_r(x_r, \theta_t) := H_t^{-1} \nabla L_r(x_r, \theta_t)$, and the first-order change along an update $\Delta\theta$ is $\Delta^{(1)} L_r = \langle \nabla^{H_t} L_r(x_r, \theta_t), \Delta\theta \rangle_{H_t}$. We omit the subscript t on H_t when the iteration is clear.

This choice is consequential for projection. Let $a_f = \nabla L_f$ and $a_r = \nabla L_r$ denote raw gradients. A Euclidean raw-gradient projector may produce $\hat{a}_f$ with $a_r^{\top} \hat{a}_f = 0$, but the optimizer applies $-\eta P_t \hat{a}_f$,

¹This variable-metric view is standard: adaptive methods such as AdaGrad and Adam act as diagonal preconditioners and thus endow a stepwise SPD metric (see, e.g., Duchi et al. (2011); Kingma and Ba (2014); natural-gradient/K-FAC metrics (Amari et al., 2019; Martens and Grosse, 2015)). Momentum and bias correction in the practical Adam implementation are discussed in Appendix B.

whose retain effect is

$$-\eta a_r^\top P_t \hat{a}_f, \quad a_r^\top \hat{a}_f = 0 \not\Rightarrow a_r^\top P_t \hat{a}_f = 0. \quad (4)$$

GU instead decomposes the *candidate update directions* $d_f = P_t a_f$ against $d_r = P_t a_r$ under $H_t = P_t^{-1}$. Thus $\langle d_r, d_f^\perp \rangle_{H_t} = 0$ is exactly $a_r^\top d_f^\perp = 0$, the first-order safety condition for the corresponding locally preconditioned displacement. This is the central difference from Euclidean raw-gradient orthogonalization. Our objective is to reduce the harm in Eq. 3, ideally keeping L_r unchanged; locally, this means enforcing $\Delta^{(1)} L_r = 0$:

Proposition 3.1. *Formally, fix $\theta \in \mathbb{R}^p$ and an optimizer-induced symmetric positive definite metric $H \succ 0$ with inner product $\langle u, v \rangle_H := u^\top H v$ and its norm is $\|v\|_H := \sqrt{\langle v, v \rangle_H}$. For each retain sample $x_r \in D_r$, assume $\ell_r(x_r; \theta)$ is differentiable and define the metric retain gradient $g(x_r) := \nabla_\theta^H \ell_r(x_r; \theta) \in \mathbb{R}^p$. Let the parameter-dependent retain gradient subspace be*

$$T_r(\theta) := \text{span}\{g(x_r) : x_r \in D_r\} \subseteq \mathbb{R}^p,$$

and its H -orthogonal complement $T_r(\theta)^\perp := \{v \in \mathbb{R}^p : \langle v, g \rangle_H = 0 \ \forall g \in T_r(\theta)\}$. For any finite collection $(x_{r_i})_{i=1}^m \subset D_r$, define the finite retain loss $L_r(\theta) := \sum_{i=1}^m \ell_r(x_{r_i}; \theta)$, $\nabla_\theta^H L_r(\theta) = \sum_{i=1}^m g(x_{r_i}) \in T_r(\theta)$. Then for any update direction $\Delta\theta \in \mathbb{R}^p$, the following are equivalent:

(i) $\Delta\theta \in T_r(\theta)^\perp$. (ii) $\Delta^{(1)} L_r(\theta; \Delta\theta) := \langle \nabla_\theta^H L_r(\theta), \Delta\theta \rangle_H = 0$ for every finite retain batch.

i.e., when L_r is locally invariant, $\Delta^{(1)} L_r = 0$, if and only if the update direction $\Delta\theta$ is H -orthogonal to T_r . This identifies $T_r^\perp$ as a *retain-invariance* subspace for forgetting. For any candidate forget descent direction s_t (whose parameter step is $-\rho s_t$), GU is equivalently the minimum-change retain-safe correction

$$\text{GU}_H(s_t) := \arg \min_{d \in T_r^\perp} \frac{1}{2} \|d - s_t\|_H^2 = P_\perp^{(H)} s_t. \quad (5)$$

The null constraint $\nabla_\theta \ell_r^\top \Delta\theta = 0$ is metric-independent, while H determines which feasible direction is closest to the candidate update and steepest under the optimizer's local trust region. Appendix C.1 proves the equivalence; Eq. 5 is the standard H -orthogonal projection identity.

Geometric Decomposition. As established above, if an update direction is H -orthogonal to the retain-gradient subspace, then the retain loss

is locally unchanged. This motivates a geometric decomposition (Fig. 1) of a forget gradient

$$g_f(x_f) = P_{T_r}^{(H)} g_f(x_f) + P_\perp^{(H)} g_f(x_f), \quad (6)$$

into a *tangential* component $P_{T_r}^{(H)} g_f(x_f) \in T_r$ and a *normal* component $P_\perp^{(H)} g_f(x_f) \in T_r^\perp$, where

$$\begin{aligned} P_{T_r}^{(H)} &= U(U^\top H U)^{-1} U^\top H, \\ P_\perp^{(H)} &= I - P_{T_r}^{(H)}. \end{aligned} \quad (7)$$

where, ideally, $U = [u_1, \dots, u_k] \in \mathbb{R}^{p \times k}$ is an H -orthonormal basis of the full retain-gradient subspace T_r . In practice, GU builds $\hat{U}$ from a small retain mini-batch on selected tensors, defining an estimate $\hat{T}_r = \text{range}(\hat{U})$ and the corresponding projectors by replacing (U, T_r) with $(\hat{U}, \hat{T}_r)$ in Eq. 7. Thus the normal component has exactly zero first-order effect on the estimated span; its residual effect relative to the ideal full-retain span is controlled by the subspace-estimation error (Appendix C.2). The identity in $P_\perp^{(H)} = I - P_{T_r}^{(H)}$ has the ambient parameter dimension. We therefore measure ideal retain-forget entanglement by

$$\text{ent}_H(g_f(x_f)) := \|P_{T_r}^{(H)} g_f(x_f)\|_H, \quad (8)$$

which vanishes iff $g_f(x_f) \in T_r^\perp$. Since $T_r^\perp$ contains many safe directions, we select the steepest feasible forget direction under a fixed H -budget. Fix the metric $H \succ 0$, First-order linearization of the joint objective $\mathcal{L}_{\text{joint}}(\theta) := L_f(\theta) + \alpha L_r(\theta)$ at θ_t gives $\Delta^{(1)} \mathcal{L}_{\text{joint}}(\theta_t; \Delta\theta) = \langle \nabla^H L_f(\theta_t) + \alpha \nabla^H L_r(\theta_t), \Delta\theta \rangle_H$. Because $\nabla^H L_r(\theta_t) \in T_r$ and $\Delta\theta \in T_r^\perp$, the retain term vanishes: $\langle \nabla^H L_r(\theta_t), \Delta\theta \rangle_H = 0$. Hence, within the retain-invariance set, optimizing the joint objective reduces to optimizing the projected forget gradient:

Lemma 3.2 (Steepest feasible descent under first-order safety). *Let $H \succ 0$ and let $T_r = \text{range}(U)$ be the retain-gradient subspace (with respect to the H -inner product). Define the feasible set*

$$\begin{aligned} \mathcal{C} &:= \{\Delta\theta \in \mathbb{R}^p : U^\top H \Delta\theta = 0, \|\Delta\theta\|_H \leq 1\} \\ &= \{v \in T_r^\perp : \|v\|_H \leq 1\}. \end{aligned}$$

For $g_f := \nabla_\theta^H L_f(\theta_t)$, the direction achieving the largest first-order decrease of L_f over $\mathcal{C}$ is

$$\Delta\theta_f^* = \arg \min_{\Delta\theta \in \mathcal{C}} \langle g_f, \Delta\theta \rangle_H = -\frac{P_\perp^{(H)} g_f}{\|P_\perp^{(H)} g_f\|_H},$$

$\Delta\theta_f^*$ is unique if $P_{\perp}^{(H)}g_f \neq 0$. Moreover, letting $g_r := \nabla_{\theta}^H L_r(\theta_t) \in T_r$, the same $\Delta\theta_f^*$ also achieves the largest first-order decrease of the joint objective $\mathcal{L}_{\text{joint}} := L_f + \alpha L_r$ over $\mathcal{C}$:

$$\Delta\theta_f^* = \arg \min_{\Delta\theta \in \mathcal{C}} \langle g_f + \alpha g_r, \Delta\theta \rangle_H.$$

The proof is in Appendix C.2. Combining the retain-invariant forget branch with an optional retain-repair branch gives:

$$\begin{aligned} \theta_{t+1} &= \theta_t - \rho(d_f^{\perp} + \beta d_r^{\parallel}), \\ d_f^{\perp} &:= P_{\perp}^{(H)} \nabla^H L_f(\theta_t), \\ d_r^{\parallel} &:= P_{T_r}^{(H)} \nabla^H L_r(\theta_t). \end{aligned} \quad (9)$$

Here $d_f^{\perp}$ is retain-orthogonal, $d_r^{\parallel}$ is retain-tangent, and $\beta \geq 0$ controls the optional retain-repair branch; $\beta = 0$ gives the pure retain-invariant projection. Under standard H -smoothness, the step size ρ can be selected by a trust-region or line-search rule (see Proposition 3.3 and Appendix C.2). In implementation, GU uses a conservative extension: it removes harmful retain-tangential forget components, but may keep sign-aware tangential components that locally decrease L_r , together with the retain-repair term. Thus the practical guarantee is first-order retain non-increase rather than exact zero drift; setting these optional terms to zero recovers the pure retain-invariant projection. Our geometric disentanglement update is plug-and-play: the projection touches only selected trainable tensors, making GU architecture-agnostic. We present Algorithm 1 in Appendix B, including basis construction and the optimizer update step. **Does projection weaken forgetting? No:** GU removes only the retain-interfering component of the forget update; unless the forget gradient lies entirely in the retain-gradient subspace, the projected direction remains a strict first-order descent direction for the forget loss (see Appendix D.3).

Theoretical Guarantees We summarize GU’s guarantee in one core statement and defer the detailed derivation to Appendix C.2.

Proposition 3.3 (Core guarantee of GU). *Let $H \succ 0$ define the optimizer-induced inner product, and let T_r be the retain-gradient subspace with projectors $P_{T_r}^{(H)}$ and $P_{\perp}^{(H)}$. For $g_f := \nabla_{\theta}^H L_f(\theta_t)$ and $g_r := \nabla_{\theta}^H L_r(\theta_t) \in T_r$, consider one GU step*

$$\Delta\theta = -\rho \left(P_{\perp}^{(H)} g_f + \beta P_{T_r}^{(H)} g_r \right), \rho > 0, \beta \geq 0. \quad (10)$$

Then GU is first-order safe on the retain loss:

$$\Delta^{(1)} L_r = \langle g_r, \Delta\theta \rangle_H = -\rho\beta \|g_r\|_H^2 \leq 0. \quad (11)$$

Moreover, if $\nabla_{\theta}^H L_r$ is $L_r^{(H)}$ -Lipschitz under $\|\cdot\|_H$, then

$$\begin{aligned} L_r(\theta_t + \Delta\theta) &\leq L_r(\theta_t) - \rho\beta \|g_r\|_H^2 \\ &+ \frac{L_r^{(H)}}{2} \rho^2 \left(\|P_{\perp}^{(H)} g_f\|_H^2 + \beta^2 \|g_r\|_H^2 \right). \end{aligned} \quad (12)$$

Thus the projected forget branch causes $O(\rho^2)$ retain drift when $\beta = 0$, while for $\beta > 0$ and a sufficiently small step size it strictly decreases L_r whenever $g_r \neq 0$. Among all unit-budget directions in $T_r^{\perp}$, the direction proportional to $-P_{\perp}^{(H)} g_f$ is the steepest first-order descent direction for forgetting.

The proposition captures the main intuition: GU forgets along retain-normal directions, repairs along retain-tangent directions, and controls finite-step retain drift through the optimizer geometry. Appendix C.2 also gives the one-step joint-objective identity and sufficient nonpositivity conditions.

4 Experiments

4.1 Settings

Datasets We evaluate on the **OpenUnlearning** suite, primarily **TOFU** (Dorna et al., 2025), a fine-grained benchmark with 200 fictitious author profiles, each containing 20 QA pairs. We use the suite’s Llama-3 backbones (1B, 3B, 8B) (Grattafiori et al., 2024) and official scaling splits (forget01/05/10) to test different forget-set sizes. We also report **MUSE** (Shi et al., 2025), which evaluates memorization and unlearning of books and news articles through verbatim reproduction, question answering, and membership inference, and **WMDP-cyber** (Li et al., 2024), an alignment-oriented benchmark assessing whether models can forget hazardous cyber capabilities while retaining general performance. For MUSE and WMDP, we use the OpenUnlearning-recommended Llama-2-7B (Touvron et al., 2023b) and zephyr-7b (Tunstall et al., 2024) backbones to ensure the target knowledge has been learned before unlearning.

Evaluation Metrics Following Dorna et al. (2025); Yang et al. (2025), we evaluate unlearning

along four axes. **Forgetting** is measured by forget-set *Extraction Strength* (Carlini et al., 2021) (ES Un.; ↓), which quantifies residual regurgitation by testing how easily the model can reconstruct target facts under constrained prompts, directly probing whether the intended knowledge has been removed. **Retention** is assessed by retain-set *Extraction Strength* (ES Re.; ↑), monitoring collateral damage to preserved knowledge; for MUSE and WMDP we additionally report *ROUGE RE*, which measures generation quality on retained QA pairs. **Privacy** captures resistance to membership inference: on TOFU we report *MIA closeness* (↑), the similarity between unlearned and retain-only models across MIA variants, while on MUSE and WMDP we use signed *Privacy Leakage* (Priv. Leak.; closer to 0 is better, equivalently |Priv. Leak.| ↓). Finally, **Utility** (↑) captures post-unlearning usefulness: TOFU uses a composite model-utility score combining probability, ROUGE, and Truth Ratio across retain and factual knowledge sets, while MUSE and WMDP report ROUGE on retained QA tasks. Together, these axes disentangle *what was forgotten* (ES Un.), *what was preserved* (ES Re., ROUGE RE), *whether leakage is controlled* (MIA, Priv. Leak.), and *whether the model remains useful* (Utility); see open-unlearning for details (Dorna et al., 2025).

4.2 TOFU Benchmark

Table 1 reports results on the TOFU benchmark. We use two informative references. *Vanilla* is the pretrained backbone without any TOFU fine-tuning; it neither learns nor regurgitates TOFU facts (low ES on both splits), enjoys perfect MIA-closeness (Priv = 1), and yields moderate utility. *Fully-finetuned* is trained on the entire TOFU corpus; it memorizes broadly (high ES on both splits), collapses privacy (Priv = 0), and reaches a utility ceiling. The practical goal is to move unlearning methods away from these two extremes toward a frontier that combines low ES on the forget set with high ES on the retain set and high utility, while keeping privacy nontrivial.

Main TOFU findings. Table 1 and Figure 2 show that GU shifts baselines toward a better aggregate retain-forget frontier. Across 72 model-split-objective configurations, GU improves or preserves ES Re. in 66/72 (91.7%), ES Un. in 51/72 (70.8%), privacy in 49/72 (68.1%), and MU in 62/72 (86.1%); no configuration worsens all

four axes. This is nontrivial because TOFU tests whether a method can suppress memorized author facts without broadly damaging nearby retained knowledge. GU tends to improve retention while maintaining competitive forgetting, rather than reducing ES Un. through indiscriminate degradation. These results support the proposed mechanism: removing harmful retain-tangential components reduces collateral damage while preserving the retain-orthogonal forgetting signal. Detailed analyses are provided in Appendix D.4.

4.3 MUSE & WMDP Benchmarks

We further evaluate GU beyond TOFU on two complementary settings. *MUSE* (Llama-2-7B) probes memorization over Books/News, where successful unlearning should *reduce* forget-set ES while *preserving* retained-QA ROUGE and moving signed privacy leakage closer to zero. *WMDP-cyber* (zephyr-7b-beta) probes capability removal, where lower unlearning accuracy (Un. acc. ↓) indicates reduced hazardous cyber capability and MMLU ↑ measures preserved general ability. Together, these benchmarks test whether GU transfers from synthetic fact removal to broader memorization and capability-unlearning settings. Table 2 shows that GU also improves the frontier in these settings, suggesting that retain-gradient projection is not specific to TOFU or to a single objective. The Pareto analysis is in Appendix D.5.

4.4 Controlled Analysis of Geometry and Composition

Table 3 directly compares GU with OrthoGrad (Shamsian et al., 2025) and UAM (Kim et al., 2025); implementation details are in Appendix E.1. The OrthoGrad study fixes the model, SimNPO objective, selected tensors, data, schedule, and evaluation. Under Euclidean geometry, GU’s streaming basis is comparable to per-sample QR while reducing runtime from 1,193s to 229s. Replacing only the metric in GU-Euclidean with the optimizer-induced metric then improves all four evaluation axes. Full GU outperforms the OrthoGrad-style baseline on every metric at $5.1 \times$ lower runtime.

The UAM study uses a controlled Cyber-only Zephyr-7B pipeline because the released UAM repository does not contain the paper’s LLM training code. At the paper learning rate, adding GU leaves UAM nearly unchanged. At the more aggressive 10^{-4} rate, UAM obtains stronger removal but

Table 1: TOFU results comparing unlearning objectives with GU across model scales and deletion rates. $\uparrow$ / $\downarrow$ indicate that larger/smaller values are better. Each “w. GU” row is compared only with its corresponding base method. Cells are shaded when adding GU improves the displayed metric: **blue** denotes an increase in a higher-is-better metric, while **pink** denotes a decrease in a lower-is-better metric. Darker shading indicates a larger improvement; no boldface is used. ES Re. and ES Un. denote extraction strength on the retain and forget splits, respectively; Priv. denotes MIA closeness; and MU denotes composite model utility. “Forget-1%, 5%, 10%” indicate the fractions of TOFU authors deleted. “Vanilla” denotes the pretrained backbone without TOFU fine-tuning, and “fully-finetuned” denotes training on the full TOFU corpus.

Method	Forget-1%				Forget-5%				Forget-10%			
	ES Re. $\uparrow$	ES Un. $\downarrow$	Priv. $\uparrow$	MU $\uparrow$	ES Re. $\uparrow$	ES Un. $\downarrow$	Priv. $\uparrow$	MU $\uparrow$	ES Re. $\uparrow$	ES Un. $\downarrow$	Priv. $\uparrow$	MU $\uparrow$
Llama-3.2-1B-Instruct												
Vanilla	0.0657	0.0692	1.0	0.5986	0.0667	0.0634	1.0	0.5991	0.0672	0.0589	1.0	0.5911
fully-finetuned	0.6483	0.7431	0.0	0.5991	0.6547	0.7271	0.0	0.5991	0.6475	0.7062	0.0	0.5991
GradDiff	0.1347	0.0410	0.6478	0.4170	0.2024	0.0327	0.6619	0.5232	0.1202	0.0325	0.5576	0.4763
GradDiff w. GU	0.1558	0.0421	0.6598	0.4417	0.2125	0.0327	0.6661	0.5308	0.1531	0.0325	0.5897	0.4798
CEU	0.0875	0.0316	0.5328	0.3666	0.0348	0.0327	0.8855	0.0000	0.0348	0.0325	0.9022	0.0000
CEU w. GU	0.2236	0.0328	0.5121	0.5134	0.2798	0.0333	0.6986	0.5635	0.4366	0.0325	0.6598	0.5844
DPO	0.3391	0.1520	0.5788	0.5071	0.2114	0.1507	0.5065	0.0710	0.2629	0.1826	0.4412	0.2157
DPO w. GU	0.3440	0.1545	0.5813	0.5099	0.2243	0.1535	0.5020	0.0922	0.2792	0.1822	0.4411	0.3016
NPO	0.3071	0.0637	0.7989	0.5482	0.1321	0.0678	0.8954	0.4378	0.1924	0.0742	0.9491	0.5218
NPO w. GU	0.3574	0.0670	0.9595	0.5520	0.1191	0.0632	0.9651	0.4623	0.2226	0.0864	0.9172	0.5442
SatImp	0.6437	0.6183	0.5112	0.5889	0.4948	0.4604	0.3591	0.5682	0.4841	0.4184	0.3804	0.5760
SatImp w. GU	0.6517	0.4855	0.5114	0.5942	0.5494	0.3964	0.3632	0.5724	0.5423	0.3459	0.3850	0.5790
SimNPO	0.6341	0.2824	0.5482	0.5899	0.4868	0.2072	0.4089	0.5696	0.4636	0.1838	0.4178	0.5781
SimNPO w. GU	0.6260	0.1204	0.7414	0.5954	0.5272	0.1140	0.6540	0.5770	0.5350	0.1099	0.6163	0.5884
UNDIAL	0.3462	0.0539	0.7994	0.5512	0.2391	0.0524	0.5697	0.5567	0.2631	0.0463	0.5246	0.5645
UNDIAL w. GU	0.5900	0.0565	0.7962	0.5886	0.6613	0.0458	0.5888	0.5972	0.6888	0.0395	0.5889	0.6026
WGA	0.5455	0.0516	0.9194	0.5872	0.4891	0.0335	0.7152	0.5836	0.4474	0.0325	0.6685	0.5825
WGA w. GU	0.6180	0.0884	0.9119	0.5963	0.5212	0.0377	0.7168	0.5773	0.4828	0.0325	0.6752	0.5862
Llama-3.2-3B-Instruct												
Vanilla	0.0689	0.0647	1.0	0.6649	0.0694	0.0656	1.0	0.6594	0.0645	0.0665	1.0	0.6623
fully-finetuned	0.8763	0.9201	0.0	0.6660	0.8459	0.8869	0.0	0.6660	0.8730	0.8904	0.0	0.6660
GradDiff	0.1241	0.0425	0.6712	0.3635	0.2273	0.0327	0.5974	0.5031	0.1808	0.0325	0.6340	0.5720
GradDiff w. GU	0.2578	0.0436	0.6625	0.5677	0.3123	0.0327	0.6145	0.5822	0.2074	0.0325	0.5842	0.6041
CEU	0.1692	0.0297	0.4265	0.5585	0.0348	0.0327	0.8963	0.0000	0.0348	0.0325	0.8627	0.0000
CEU w. GU	0.3046	0.0291	0.4288	0.6159	0.3411	0.0332	0.6924	0.6255	0.5568	0.0325	0.6350	0.6672
DPO	0.5017	0.3143	0.6057	0.6273	0.3098	0.1973	0.4564	0.1023	0.3866	0.2598	0.4264	0.3283
DPO w. GU	0.5035	0.2569	0.6097	0.6266	0.3233	0.2058	0.4544	0.1867	0.4104	0.2564	0.4270	0.4526
NPO	0.4129	0.0865	0.7520	0.6356	0.1454	0.0595	0.8653	0.4828	0.1389	0.0600	0.8746	0.5329
NPO w. GU	0.4941	0.0947	0.9232	0.6499	0.1329	0.0632	0.9526	0.4326	0.1936	0.0704	0.9604	0.5805
SatImp	0.7926	0.7171	0.5836	0.6429	0.6147	0.6210	0.3546	0.6370	0.5739	0.5426	0.3884	0.6457
SatImp w. GU	0.8134	0.5973	0.5852	0.6480	0.6529	0.4744	0.3598	0.6441	0.6200	0.4386	0.3934	0.6362
SimNPO	0.7490	0.3896	0.6349	0.6417	0.6068	0.2470	0.4046	0.6342	0.5682	0.2032	0.4511	0.6439
SimNPO w. GU	0.7781	0.1747	0.7842	0.6447	0.7977	0.1089	0.6467	0.6670	0.6224	0.1207	0.6745	0.6488
UNDIAL	0.4396	0.0658	0.8748	0.6468	0.3242	0.0465	0.6397	0.6463	0.3538	0.0416	0.5833	0.6550
UNDIAL w. GU	0.6996	0.0619	0.8736	0.6805	0.7641	0.0424	0.6616	0.6935	0.7869	0.0396	0.6226	0.6992
WGA	0.6827	0.0818	0.9365	0.6522	0.6060	0.0327	0.6975	0.6417	0.6427	0.0341	0.6516	0.6497
WGA w. GU	0.7440	0.1226	0.9425	0.6543	0.6163	0.0327	0.6751	0.6419	0.6425	0.0334	0.6544	0.6451
Llama-3.1-8B-Instruct												
Vanilla	0.0674	0.0645	1.0	0.6176	0.0697	0.0741	1.0	0.6322	0.0645	0.0650	1.0	0.6461
fully-finetuned	0.9247	0.9767	0.0	0.6276	0.9238	0.9719	0.0	0.6276	0.9463	0.9789	0.0	0.6276
GradDiff	0.3072	0.0764	0.5497	0.5481	0.2897	0.0327	0.5666	0.5890	0.3098	0.0325	0.5638	0.5713
GradDiff w. GU	0.3449	0.0756	0.5475	0.5659	0.4639	0.0327	0.6402	0.6276	0.3408	0.0325	0.6233	0.5771
CEU	0.1500	0.0291	0.5311	0.5465	0.0348	0.0327	0.9180	0.0000	0.0348	0.0325	0.8689	0.0000
CEU w. GU	0.3050	0.0291	0.5091	0.6144	0.4470	0.0327	0.6370	0.6411	0.6987	0.0325	0.6856	0.6773
DPO	0.5852	0.2854	0.5593	0.5774	0.4902	0.2476	0.4607	0.2390	0.7038	0.3366	0.4451	0.3281
DPO w. GU	0.5840	0.2445	0.5702	0.5813	0.5512	0.2329	0.4661	0.3523	0.7557	0.3188	0.4565	0.4481
NPO	0.3861	0.0811	0.7948	0.5717	0.2071	0.0648	0.7440	0.5839	0.2435	0.0684	0.7516	0.6104
NPO w. GU	0.4006	0.0818	0.7972	0.5842	0.2642	0.0664	0.7447	0.6173	0.3476	0.0708	0.7396	0.6095
SatImp	0.9505	0.9037	0.5186	0.6269	0.7967	0.7391	0.3641	0.6013	0.6794	0.6436	0.3846	0.6265
SatImp w. GU	0.9342	0.7251	0.5248	0.6326	0.8120	0.4872	0.3782	0.6163	0.6978	0.4360	0.3891	0.6118
SimNPO	0.8256	0.3101	0.6178	0.6270	0.7814	0.2529	0.4762	0.6040	0.6530	0.2110	0.4789	0.6029
SimNPO w. GU	0.8284	0.1177	0.7810	0.6269	0.8067	0.1425	0.7424	0.6227	0.7103	0.1140	0.7027	0.6519
UNDIAL	0.5679	0.0683	0.7964	0.7089	0.5453	0.0506	0.5812	0.6781	0.6213	0.0495	0.5364	0.6934
UNDIAL w. GU	0.7520	0.0671	0.8115	0.7257	0.9191	0.0477	0.6029	0.6860	0.9349	0.0468	0.5361	0.6802
WGA	0.7644	0.0745	0.7161	0.6258	0.7299	0.0339	0.6199	0.6256	0.6560	0.0339	0.6552	0.6331
WGA w. GU	0.7915	0.0785	0.7513	0.6297	0.7577	0.0327	0.6184	0.6425	0.7122	0.0325	0.6225	0.6024

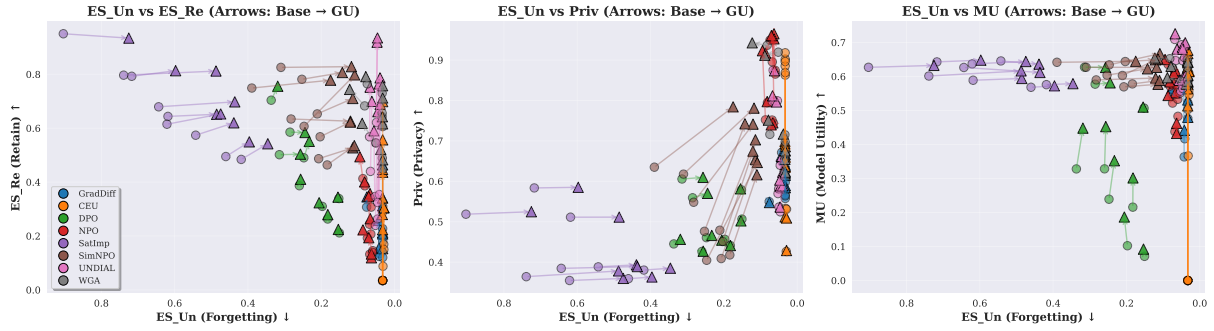


Figure 2: We visualize forgetting quality (ES Un.; lower is better) against retained knowledge (ES Re.), privacy (Priv.), and model utility (MU) for eight unlearning baselines on TOFU. The ES Un. axis is reversed, so moving right means lower ES Un.; ES Re., Priv., and MU are higher-is-better metrics. Circles denote baseline outputs, triangles denote GU results, and arrows indicate the shift from Base $\rightarrow$ GU. Across the panels, GU usually moves methods toward a better retain-forget frontier.

loses 0.3206 MMLU relative to Base; GU recovers 0.3011 (94%) of this loss, while WMDP-Cyber accuracy remains below Base. This is a safety-forgetting trade-off rather than uniform dominance: GU weakens the aggressive removal from 0.2506 to 0.4057 while preventing utility collapse, consistent with its role as a retain-safety controller.

4.5 Ablations and Cost Analysis

We summarize the main ablations below, with full results in Appendix E.

Optimizer-induced geometry. Replacing the Euclidean metric with the Adam-induced update metric improves SimNPO+GU on both Llama-3.2-1B and 3B, reducing ES Un. from 0.26 to 0.21 and from 0.21 to 0.17, respectively, while also improving ES Re., privacy, and MU.

Optimizer compatibility. GU improves SimNPO under both AdamW and SGD on Llama-3.2-3B Forget01. Under AdamW, ES Un. decreases from 0.39 to 0.17 and privacy increases from 0.63 to 0.78; under SGD, they improve from 0.42 to

Table 2: MUSE benchmark results for Llama-2-7b-hf on the Books and News splits and WMDP benchmark results for zephyr-7b-beta on the cyber split. ↓ indicates that smaller values are better, ↑ indicates that larger values are better, and $\rightarrow 0$ indicates that the signed privacy leakage should move closer to zero. Each “w. GU” row is compared only with its corresponding base method. Cells are shaded when adding GU improves the displayed metric: pink denotes a decrease in a lower-is-better metric, while blue denotes an increase in a higher-is-better metric or a reduction in the absolute signed privacy leakage. Darker shading indicates a larger improvement after accounting for metric scale. “Vanilla” denotes the pretrained backbone without benchmark-specific fine-tuning, and “fully-finetuned” denotes training on the full benchmark before unlearning.

Method	MUSE Books			MUSE News			WMDP cyber	
	ES Un. ↓	Priv. Leak. $\rightarrow 0$	ROUGE Re. ↑	ES Un. ↓	Priv. Leak. $\rightarrow 0$	ROUGE Re. ↑	Un. acc. ↓	MMLU acc. ↑
Llama-2-7b-hf								
Vanilla	0.01	8.16	0.68	0.02	-4.72	0.56	0.4453	0.5845
fully-finetuned	0.92	-57.34	0.69	0.29	-99.81	0.55	-	-
GD	0.0079	-24.5562	0.0	0.0116	88.2242	0.3971	0.2420	0.4772
GD w. GU	0.0079	-24.6394	0.0	0.0085	88.0562	0.3992	0.2375	0.4937
CEU	0.0079	-58.8018	0.0	0.0079	-7.3468	0.0	0.2455	0.2689
CEU w. GU	0.0079	-58.0251	0.0	0.0182	66.1418	0.4349	0.2455	0.2689
NPO	0.3933	-54.4933	0.6185	0.1021	-85.8312	0.5050	0.3457	0.5422
NPO w. GU	0.3822	-53.7352	0.6251	0.1175	-86.04	0.5037	0.3668	0.5518
SatImp	0.7710	-58.3950	0.6114	0.2287	-99.8741	0.3991	0.4177	0.5654
SatImp w. GU	0.7321	-57.3851	0.6310	0.1943	-99.8740	0.4100	0.4157	0.5674
SimNPO	0.1407	-54.2530	0.5103	0.1778	-99.8741	0.4114	0.4192	0.5658
SimNPO w. GU	0.0813	-46.4866	0.5980	0.0957	-99.8740	0.4143	0.4177	0.5663
UNDIAL	0.0231	-18.3432	0.6309	0.0110	-98.9085	0.1928	0.3829	0.5596
UNDIAL w. GU	0.0219	-18.2137	0.6370	0.0168	-99.37	0.3638	0.3789	0.5612
WGA	0.0079	-49.9445	0.4689	0.0102	101.1335	0.4602	0.2455	0.2550
WGA w. GU	0.0079	-40.1072	0.4682	0.0084	108.14	0.4615	0.3819	0.5498

Table 3: Controlled analyses. Panel (b) reports method $\rightarrow$ method+GU; bold marks the preferred value.

(a) Projection geometry (TOFU Forget01, Llama-3.2-B)								(b) Method composition (WMDP-Cyber)			
Method	Basis	Metric	ES Re. $\uparrow$	ES Un. $\downarrow$	Priv. $\uparrow$	MU $\uparrow$	Time $\downarrow$	Method	LR	WMDP $\downarrow$	MMLU $\uparrow$
OrthoGrad-style	Sample QR	Euclid.	0.76	0.22	0.77	0.63	1,193	RMU	-	0.2416 $\rightarrow^{GU}$ 0.2406	0.5763 $\rightarrow^{GU}$ 0.5784
GU-Euclidean	Stream- k	Euclid.	0.76	0.21	0.76	0.64	229	UAM	5×10^{-5}	0.4283 $\rightarrow^{GU}$ 0.4287	0.5830 $\rightarrow^{GU}$ 0.5837
GU	Stream- k	Optim.	0.78	0.17	0.78	0.65	233	UAM	10^{-4}	0.2506 $\rightarrow^{GU}$ 0.4057	0.2648 $\rightarrow^{GU}$ 0.5659

0.20 and from 0.61 to 0.75, respectively.

Seed stability. Across random seeds, the standard deviation is approximately 0.01 for all metrics, substantially smaller than the primary improvements in ES Un. (-0.22) and privacy ($+0.15$).

Computational cost. On Llama-3.2-1B, GU adds only 1.3-3.0% wall-clock and less than 5% peak-memory overhead. A projection rank of $k = 8$ already matches larger ranks.

5 Related Work

LLM Unlearning LLM unlearning modifies a trained model to remove a forget set while approximating retain-only retraining (Dorna et al., 2025). Gradient ascent with retain regularization can catastrophically forget (Thudi et al., 2022; Izzo et al., 2021); NPO/SimNPO/DPO and calibration-based variants improve stability by reshaping forget updates (Zhang et al., 2024a; Fan et al., 2025; Rafailov et al., 2023; Dong et al., 2025). Other approaches manipulate representations, distill behavior, or mask information at inference time (Wang et al., 2025b; Ginart et al., 2019; Le-Khac and Truong, 2025; Liu et al., 2024). Yet target content can remain extractable and behavior can differ from retain-only retraining (Maini et al., 2024; Li et al., 2024; Dorna et al., 2025), motivating explicit control of forget–retain interference.

Geometric Control via Orthogonal Projection

Projection methods control interference by restricting updates relative to protected gradients or subspaces (Kirkpatrick et al., 2017; Yu et al., 2020; Farajtabar et al., 2020). In unlearning, this includes common-descent alignment (Ko et al., 2024), per-sample gradient projection (OrthoGrad (Shamsian et al., 2025)), cosine regularization (Mandal and Gottwald, 2025), min-norm constraints (Block et al., 2025), and representation/statistics subspaces (Feng et al., 2025; Hoang et al., 2024; Chen et al., 2024; Sendera et al., 2025). UAM (Kim et al., 2025) instead optimizes retain loss at forget-perturbed weights, introducing a Euclidean forget-

Table 4: Positioning against the closest geometric methods by modified object, retain-safety interpretation, and projection geometry.

Method	Modified object	Retain-safety link	Geometry
Ko et al.	Joint update	Joint-objective improvement	Euclidean
UAM	Retain update	Loss at forget-perturbed weights	Euclidean
OrthoGrad	Forget gradient	Raw-gradient orthogonality	Euclidean
PGU/UNSC	Update/statistics	Proxy-structure preservation	Euclidean/SVD
GU	Precond. update	First-order retain invariance	Optimizer-induced

direction projector in its approximate outer gradient.

GU’s novelty lies in three coupled choices summarized in Table 4: the protected object is the retain-gradient span derived from first-order invariance; projection acts on a *preconditioned candidate direction*, rather than only the raw gradient; and orthogonality is defined by $H = P^{-1}$, the geometry induced by a locally frozen optimizer preconditioner. Consequently, Euclidean orthogonality can be broken by preconditioner scaling (Eq. 4), whereas GU constrains the corresponding candidate displacement. Its streaming rank- k construction also avoids OrthoGrad’s per-sample QR and the d^2 factorizations used by statistics-based methods. Table 3 isolates these differences empirically; Appendix A.3 gives details.

6 Conclusion

We studied when forgetting updates leave retained knowledge unchanged and showed that local retain-invariance corresponds to orthogonality to the retain-gradient subspace under the optimizer-induced update metric. Building on this characterization, Geometric Disentanglement Unlearning (GU) projects candidate forget updates onto the retain-orthogonal complement; in practice, it can also add retain-repair and sign-aware retain-helpful components. GU is plug-and-play, optimizer-compatible, and architecture-agnostic, and it attaches to existing gradient-based unlearning pipelines without changing their objectives. Across TOFU, MUSE, and WMDP, GU improves the aggregate retain-forget frontier and utility, while often improving forgetting and privacy.

Limitations

GU provides a local, first-order retain-safety guarantee rather than a global guarantee that all retained behavior is unchanged after many nonconvex optimization steps. In practice, the retain-gradient subspace is estimated from mini-batches and selected trainable tensors, so the quality of the projection depends on the representativeness, refresh frequency, and rank of this approximation. GU also assumes white-box access to gradients and optimizer states, which makes it less directly applicable to black-box or purely inference-time unlearning settings. Finally, although our experiments cover TOFU, MUSE, and WMDP across multiple LLM backbones and objectives, broader validation is still needed for larger models, multilingual or multimodal corpora, stronger adaptive extraction attacks, and deployment settings with continuously changing forget requests.

Ethical Considerations

This work studies machine unlearning for language models, a setting with both privacy-preserving benefits and dual-use risks. Improved unlearning methods may help remove private, copyrighted, or hazardous information from deployed models, but incomplete unlearning can also create a false sense of privacy or safety if residual information remains recoverable under adaptive extraction attacks. Conversely, unlearning techniques could be misused to conceal model provenance, suppress legitimate information, or selectively remove behaviors without transparent auditing. Our experiments are conducted on public research benchmarks and report aggregate metrics only. We do not release new hazardous datasets or pretrained model weights; released code and configurations are intended for research and reproducibility. Deployment of unlearning systems should include independent auditing, stronger adaptive extraction evaluations, and monitoring for unintended degradation on retained or safety-critical capabilities.

Acknowledgments

We thank the anonymous reviewers and area chairs for their constructive feedback. This work is supported in part by NSF (2433308), IES (2229612), NSF (2416070), NSF (IIS-2331967) and the AI2050 program at Schmidt Sciences (AI2050 Early Career Fellowship).

References

- Shun-ichi Amari, Ryo Karakida, and Masafumi Oizumi. 2019. Fisher information and natural gradient learning in random deep networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 694–702. PMLR.
- Jacob L Block, Aryan Mokhtari, and Sanjay Shakkottai. 2025. Machine unlearning under overparameterization. *arXiv preprint arXiv:2505.22601*.
- Lucas Bourtole, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE.
- Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. *2015 IEEE Symposium on Security and Privacy*, pages 463–480.
- Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.
- Huiqiang Chen, Tianqing Zhu, Xin Yu, and Wanlei Zhou. 2024. [Machine unlearning via null space calibration](#). In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*.
- Jiaao Chen and Diyi Yang. 2023. [Unlearn what you want to forget: Efficient unlearning for LLMs](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12041–12052, Singapore. Association for Computational Linguistics.
- Yijiang River Dong, Hongzhou Lin, Mikhail Belkin, Ramon Huerta, and Ivan Vulić. 2025. [UNDIAL: Self-distillation with adjusted logits for robust unlearning in large language models](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8827–8840, Albuquerque, New Mexico. Association for Computational Linguistics.
- Vineeth Dorna, Anmol Reddy Mekala, Wenlong Zhao, Andrew McCallum, J Zico Kolter, Zachary Chase Lipton, and Pratyush Maini. 2025. [Openunlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

- Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. 2025. [Simplicity prevails: Rethinking negative preference optimization for LLM unlearning](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. 2020. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pages 3762–3773. PMLR.
- Qian Feng, JiaHang Tu, Mintong Kang, Hanbin Zhao, Chao Zhang, and Hui Qian. 2025. Fg-oriu: Towards better forgetting via feature-gradient orthogonality for incremental unlearning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1957–1967.
- Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. [The Llama 3 herd of models](#). *arXiv preprint arXiv:2407.21783*.
- Laura Graves, Vineel Nagisetty, and Vijay Ganesh. 2021. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11516–11524.
- Tuan Hoang, Santu Rana, Sunil Gupta, and Svetha Venkatesh. 2024. Learn to unlearn for deep neural networks: Minimizing unlearning interference with gradient projection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4819–4828.
- Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. 2021. Approximate data deletion from machine learning models. In *International conference on artificial intelligence and statistics*, pages 2008–2016. PMLR.
- Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana Kompella, Sijia Liu, and Shiyu Chang. 2024. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *Advances in Neural Information Processing Systems*, 37:12581–12611.
- Hoki Kim, Keonwoo Kim, Sungwon Chae, and Sangwon Yoon. 2025. Unlearning-aware minimization. In *Advances in Neural Information Processing Systems*, volume 38, pages 93806–93829.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.
- Myeongseob Ko, Henry Li, Zhun Wang, Jonathan Patsenker, Jiachen T. Wang, Qinbin Li, Ming Jin, Dawn Song, and Ruoxi Jia. 2024. Boosting alignment for post-unlearning text-to-image generative models. In *Advances in Neural Information Processing Systems*, volume 37, pages 85131–85154.
- Uyen N Le-Khac and Vinh NX Truong. 2025. A survey on large language models unlearning: taxonomy, evaluations, and future directions. *Artificial Intelligence Review*, 58(12):399.
- Nathaniel Li, Alexander Pan, Anjali Gopal, Sumner Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, and 27 others. 2024. [The wmdp benchmark: Measuring and reducing malicious use with unlearning](#). In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 28525–28550. PMLR.
- Chris Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. 2024. Large language model unlearning via embedding-corrupted prompts. *Advances in Neural Information Processing Systems*, 37:118198–118266.
- Zheyuan Liu, Suraj Maharjan, Fanyou Wu, Rahil Parikh, Belhassen Bayar, Srinivasan H. Sengamedu, and Meng Jiang. 2025. [Disentangling biased knowledge from reasoning in large language models via machine unlearning](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6105–6123, Vienna, Austria. Association for Computational Linguistics.
- Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. 2024. [TOFU: A task of fictitious unlearning for LLMs](#). *First Conference On Language Modeling*.
- Pinak Mandal and Georg A Gottwald. 2025. Uno: Unlearning via orthogonalization in generative models. *arXiv preprint arXiv:2506.04712*.
- James Martens and Roger Grosse. 2015. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417. PMLR.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

- Marcin Sendera, Łukasz Struski, Kamil Ksikażek, Kryspin Musiol, Jacek Tabor, and Dawid Damian Rymarczyk. 2025. [SEMU: Singular value decomposition for efficient machine unlearning](#). In *Forty-second International Conference on Machine Learning*.
- Aviv Shamsian, Eitan Shaar, Aviv Navon, Gal Chechik, and Ethan Fetaya. 2025. Go beyond your means: Unlearning with per-sample gradient orthogonalization. *arXiv preprint arXiv:2503.02312*.
- Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Mal-ladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. 2025. [MUSE: Machine unlearning six-way evaluation for language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. 2022. [Unrolling SGD: Understanding Factors Influencing Machine Unlearning](#). In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 303–319, Los Alamitos, CA, USA. IEEE Computer Society.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. [Llama: Open and efficient foundation language models](#). *Preprint*, arXiv:2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M Rush, and Thomas Wolf. 2024. [Zephyr: Direct distillation of LM alignment](#). In *First Conference on Language Modeling*.
- Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q Weinberger. 2025a. [Re-thinking LLM unlearning objectives: A gradient perspective and go beyond](#). In *The Thirteenth International Conference on Learning Representations*.
- Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Shah, Yujia Bao, Yang Liu, and Wei Wei. 2025b. [LLM unlearning via loss adjustment with only forget data](#). In *The Thirteenth International Conference on Learning Representations*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pi-er-ic Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Trans-formers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Bo Yang. 2025. CE-U: Cross Entropy unlearning. *arXiv preprint arXiv:2503.01224*.
- Puning Yang, Qizhou Wang, Zhuo Huang, Tongliang Liu, Chengqi Zhang, and Bo Han. 2025. [Exploring criteria of loss reweighting to enhance LLM unlearning](#). In *Forty-second International Conference on Machine Learning*.
- Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024. [Machine unlearning of pre-trained large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8403–8419, Bangkok, Thailand. Association for Computational Linguistics.
- Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. [Gradient surgery for multi-task learning](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc.
- Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024a. [Negative preference optimization: From catastrophic collapse to effective unlearning](#). *First Conference on Language Modelling*.
- Yuji Zhang, Sha Li, Jiateng Liu, Pengfei Yu, Yi R Fung, Jing Li, Manling Li, and Heng Ji. 2024b. Knowledge overshadowing causes amalgamated hallucination in large language models. *arXiv preprint arXiv:2407.08039*.
- Yuji Zhang, Sha Li, Cheng Qian, Jiateng Liu, Pengfei Yu, Chi Han, Yi R. Fung, Kathleen McKeown, ChengXiang Zhai, Manling Li, and Heng Ji. 2025. [The law of knowledge overshadowing: Towards understanding, predicting and preventing LLM hallucination](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23340–23358, Vienna, Austria. Association for Computational Linguistics.

Appendix

A Discussion

A.1 Discussion on Optimizer-induced Metric

We work in parameter space $\mathbb{R}^p$. Let an optimizer use a local linear preconditioner $P_t \succ 0$ so that a raw gradient g_t induces the update direction $-P_t g_t$. We define the update-space metric as

$$H_t := P_t^{-1}.$$

Under this choice, the metric gradient is $\nabla^{H_t} L = H_t^{-1} \nabla L = P_t \nabla L$, the optimizer-preconditioned candidate direction under the locally frozen P_t . For diagonal Adam,

$$P_t = \text{diag}(1/\sqrt{\hat{v}_t + \varepsilon}), \\ H_t = \text{diag}(\sqrt{\hat{v}_t + \varepsilon}),$$

where $\hat{v}_t$ is Adam’s second-moment accumulator. To implement the projector, we use the square-root whitener $C_t = H_t^{1/2} = P_t^{-1/2}$: a preconditioned update direction $d = P_t g$ is represented as $\tilde{d} = C_t d = P_t^{1/2} g$, and Euclidean projection in $\tilde{d}$ coordinates is equivalent to H_t -orthogonal projection in update space. Then local retain invariance $\Delta^{(1)} L_r = \langle \nabla^{H_t} L_r, \Delta \theta \rangle_{H_t} = 0$ is equivalent to $\Delta \theta \in T_r^\perp$ under H_t , and the first-order safety statements apply exactly to this ideal projected direction. The practical implementation injects its equivalent gradient before Adam’s first-moment update; momentum, bias correction, and the low-rank mini-batch approximation are distinguished in Appendix B.

A.2 Constructing the retain-orthogonal space.

To protect retained behavior, we here introduce how we derive T_r on D_r from a retain loss:

$$L_r(\theta) = \mathbb{E}_{x \in D_r} \left[\text{KL}(\pi_\theta(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) \right],$$

which yields low-variance and stable gradients, a zero-gradient baseline near $\pi_\theta \approx \pi_{\text{ref}}$, and alignment with preserving output style.

From a small retain mini-batch $B_r \subset D_r$ on selected tensors, we collect retain update directions to form $\hat{U}$ and orthonormalize in H -whitened coordinates (Gram–Schmidt). These directions define an estimated subspace $\hat{T}_r$, rather than the ideal full-data span T_r . For any $v \in \mathbb{R}^p$, the projection $P_{\hat{T}_r^\perp}^{(H)} v$ lies in $\hat{T}_r^\perp$ and satisfies $\hat{U}^\top H P_{\hat{T}_r^\perp}^{(H)} v = 0$.

Thus the first-order guarantee is exact on $\hat{T}_r$; the residual relative to T_r is bounded by their H -subspace discrepancy in Appendix C.2.

A.3 Discussion on Projection-based Unlearning Methods

This section expands the main-text comparison between GU and prior projection-based unlearning methods: common-descent alignment (Ko et al., 2024), UAM (Kim et al., 2025), UNSC (Chen et al., 2024), PGU (Hoang et al., 2024), SEMU (Sendera et al., 2025), OrthoGrad (Shamsian et al., 2025), UNO (Mandal and Gottwald, 2025), FG-OrIU (Feng et al., 2025), and MinNorm-OG (Block et al., 2025). These works share a geometric vocabulary but differ in the update they modify, the object they protect, and the geometry in which their constraint is defined.

The Target Unlearning Regime is Fundamentally Different.

UNSC (Chen et al., 2024) and PGU (Hoang et al., 2024) are designed for supervised image classification with a small, fixed label space, where one can meaningfully define *class-conditional* representation statistics and compute dataset-level matrices (e.g., per-class covariances or Gram matrices) to protect retain classes. SEMU (Sendera et al., 2025) similarly assumes a supervised setup and constructs low-rank subspaces from forget-set gradients/statistics to parameterize the unlearning update. In contrast, our target is large-scale *LLM unlearning in the post-training regime* (offline SFT and preference tuning), where outputs are open-vocabulary sequences, “classes” are not well-defined, and unlearning is performed *post hoc* as a stage in a multi-step alignment pipeline. Under this regime, the structural assumptions behind class-based or dataset-statistics-based projections (and the need to repeatedly recompute such statistics across stages) do not directly apply.

These Methods Project onto Different Geometric Objects.

Even within “projection” methods, the protected object varies substantially. (1) **Representation/statistics-based subspaces.** UNSC and PGU define protected directions through proxies derived from activations or dataset statistics (e.g., class-conditional activation subspaces, retain-only Gram/covariance structure), with the goal that preserving these proxies approximately preserves retain behavior. FG-OrIU (Feng et al., 2025) further moves projection to the representation level for incremental unlearning by estimating layerwise forget/retain feature subspaces (typically via SVD) and applying forward-feature and backward-gradient projections while training

low-rank adapters. (2) **Forget-dominated low-rank parameterizations.** SEMU constructs a low-rank subspace primarily from D_f -derived information and restricts updates to that subspace; retention is handled indirectly through low rank and step-size control rather than through an explicit retain-side invariance constraint. (3) **Gradient-orthogonality constraints.** OrthoGrad (Shamsian et al., 2025) enforces hard orthogonalization by estimating a retain subspace from *per-sample* retain gradients (via QR) and projecting the forget update onto its orthogonal complement. In addition, OrthoGrad orthogonalizes gradients in the *Euclidean geometry*, whereas GU orthogonalizes candidate directions in the optimizer-induced SPD geometry derived from a retain-invariance specification; the latter aligns the safety constraint with the locally preconditioned direction. UNO (Mandal and Gottwald, 2025) (for generative models) instead imposes a *soft* cosine-orthogonality regularizer between forget/retain gradients inside the objective, making the effect sensitive to penalty tuning. MinNorm-OG (Block et al., 2025) formulates unlearning as constrained optimization with an explicit feasibility constraint $\Delta \in \text{span}(g_{\text{retain}})^\perp$, selecting approximately minimum-norm updates under this constraint. (4) **Objective-aware update shaping.** Ko et al. (Ko et al., 2024) seek common update directions that monotonically improve unlearning and text–image alignment in diffusion models. UAM (Kim et al., 2025) instead perturbs the parameters along the forget gradient and minimizes the retain loss at the perturbed weights; its efficient approximation contains a Euclidean projector onto the forget-gradient direction. These approaches modify the optimization rule itself. GU is complementary because it filters the candidate correction produced by an existing objective, as demonstrated by the UAM+GU experiment in Table 3.

GU differs in the *definition* of the protected geometry: rather than using indirect proxies (activations/Grams) or relying on forget-only low-rank parameterizations, we define the retain geometry directly in parameter space as the retain-gradient span under the **optimizer-induced SPD metric** H ,

$$T_r(\theta) = \text{span}\{\nabla_\theta^H \ell_r(x_r; \theta) : x_r \in D_r\},$$

and we prove that *local retain invariance* is equivalent to H -orthogonality to $T_r(\theta)$. Within the retain-orthogonal set, the GU update is the steepest-descent direction for the chosen unlearning objec-

tive. Thus, in GU the retain-gradient subspace is the primary geometric object and is explicitly aligned with the optimizer geometry, rather than a representation-level proxy constructed from activations, Grams, or ad hoc low-rank parameterizations.

Scalability is the main practical barrier for LLM baselines. A second key distinction is computational scaling. UNSC/PGU require computing and factorizing per-class activation covariances or retain Grams (often layerwise), and FG-OrIU/SEMU require repeated layerwise SVD-style operations over high-dimensional representations or gradient-derived matrices. For LLMs with hidden dimension $d \approx 4\text{k}–8\text{k}$ and many layers, these procedures involve $d \times d$ objects across layers, leading to prohibitive memory and factorization cost, especially when unlearning must be repeated across stages on top of existing SFT/preference tuning. Any “lightweight” adaptation (e.g., collapsing outputs into pseudo-classes, dropping most layers, or aggressively subsampling statistics) would deviate substantially from the original algorithms, making negative results difficult to interpret.

GU is explicitly designed to avoid d^2 -scale computations: we never form activation/Gram matrices and never run layerwise SVDs. Instead, we maintain a low-rank retain basis $B \in \mathbb{R}^{d \times k}$ in parameter space using streaming updates from retain gradients, and project forget gradients via vector-level operations under H (e.g., $g_f^\perp = g_f - \Pi_{T_r}^H g_f$), yielding $O(kd)$ cost per step (plus small $O(k^2)$ book-keeping). This turns geometric disentanglement into a lightweight *routing layer* that introduces only modest overhead on top of existing LLM unlearning pipelines (e.g., SimNPO, SatImp, WGA, NPO), making it practically deployable at LLM scale.

Baseline scope in our LLM experiments. For UNSC/PGU/FG-OrIU/SEMU, a faithful implementation at LLM scale is either infeasible because of repeated d^2 -scale statistics/factorizations or would require approximations that change the method qualitatively. UNO targets a different generative-model objective class, and MinNorm-OG emphasizes minimum-norm constrained solutions rather than plug-and-play post-training. We therefore implement the two closest feasible controls directly: OrthoGrad-style per-sample QR on TOFU and UAM on WMDP-Cyber. Table 3 shows that GU’s streaming construction removes OrthoGrad’s

main scaling bottleneck, its optimizer geometry adds gains beyond the shared Euclidean projection mechanism, and its residual formulation composes with UAM without replacing UAM’s optimization rule.

B Algorithm Details

Here, we introduce the detailed practical implementation of Geometric Disentanglement Unlearning (GU) in Algorithm 1. At each step we only project a small, automatically selected subset of trainable tensors: the last K Transformer blocks (largest layer indices detected from names such as `.layers.i.`, `.h.i.`, `.blocks.i.`, `.decoder.layers.i.`), plus the final normalization(s) and the output head. This keeps cost and memory small while targeting the most forget-sensitive layers.

We work with optimizer-preconditioned update directions. For each selected parameter tensor p , let v_p denote Adam’s second moment estimate; then

$$\begin{aligned} P_p &= \text{diag}\left(\frac{1}{\sqrt{v_p + \varepsilon}}\right), \\ H_p &= P_p^{-1} = \text{diag}(\sqrt{v_p + \varepsilon}), \\ C_p &= H_p^{1/2} = P_p^{-1/2}. \end{aligned}$$

In practice, we *bind* to the optimizer state and reuse v_p without extra memory; if unavailable, we maintain an EMA of squared gradients. All projections are performed in H -whitened update coordinates. For a raw gradient g , the optimizer-preconditioned direction is $d = P_p g$ and its whitened representation is $\tilde{d} = C_p d = P_p^{1/2} g$. When the optimizer API expects a gradient rather than an update direction, we map a projected direction d^* back to the equivalent gradient $g^* = H_p d^*$; in the frozen-preconditioner, no-momentum model, $-P_p g^* = -d^*$. The practical Adam momentum caveat is stated below.

How we find $P_{\hat{T}_r}^{(H)}$ and $P_{\hat{T}_r^\perp}^{(H)}$. Per selected tensor p , we maintain a small basis $U_p = \{u_{p,1}, \dots, u_{p,m}\}$ (with $m \leq k$) in H_p -whitened update coordinates. The corresponding unwhitened directions span an estimate $\hat{T}_{r,p}$ of the ideal retain tangent subspace. The basis is refreshed at low frequency using one backward pass of a lightweight retain anchor

$$\mathcal{L}_r(\theta) = \mathbb{E}_{x \in D_r} \text{KL}(\pi_\theta(\cdot|x) \parallel \pi_{\text{ref}}(\cdot|x)).$$

For each p , we compute the raw gradient $g_{r,p} = \partial \mathcal{L}_r / \partial p$, form the preconditioned direction $d_{r,p} =$

Algorithm 1 Geometric Disentanglement Unlearning

- Require:** At step t , Parameters θ_t , optimizer-induced metric H_t (preconditioner P_t , whitener C_t), reference model θ_{ref} , batches B_f, B_r , weights γ, α , any forget loss function L_f and retain loss function L_r
- 1: Compute $L_f(\theta_t; B_f)$ and $L_r(\theta_t; B_r, \theta_{\text{ref}})$, form $L_{\text{tot}} = \gamma L_f + \alpha L_r$ and total gradient $g_{\text{tot}} = \nabla_\theta L_{\text{tot}}$
 - 2: Compute retain KL anchor L_r^{KL} and raw gradient g_r , form $d_r = P_t g_r$, whiten $\tilde{d}_r = C_t d_r$, and update the retain basis U_t to approximate $T_r(\theta_t)$.
 - 3: Recover raw forget gradient via $g_f = (g_{\text{tot}} - \alpha g_r) / \gamma$, form $d_f = P_t g_f$, and whiten $\tilde{d}_f = C_t d_f$
 - 4: Decompose $\tilde{d}_f$ w.r.t. U_t : obtain retain-orthogonal $\tilde{d}_f^\perp$ and a sign-selective, norm-capped tangent part $\tilde{d}_f^{\text{tan,keep}}$; similarly get $\tilde{d}_r^{\text{tan}}$
 - 5: Form whitened GU direction $\tilde{d}_{\text{GU}} = \gamma(\tilde{d}_f^\perp + \tilde{d}_f^{\text{tan,keep}}) + \alpha \tilde{d}_r^{\text{tan}}$, map back $d_{\text{GU}} = C_t^{-1} \tilde{d}_{\text{GU}}$, set the equivalent gradient $g_{\text{GU}} = H_t d_{\text{GU}}$, and let the base optimizer step with g_{GU} to obtain θ_{t+1}

Ensure: Updated parameters θ_{t+1}

$P_p g_{r,p}$, whiten $\tilde{d}_{r,p} = C_p d_{r,p} = P_p^{1/2} g_{r,p}$, and run Gram–Schmidt against the current U_p in float32; if the relative residual $\|\text{res}\| / \|\tilde{d}_{r,p}\|$ exceeds `residual_keep_thresh` and $|U_p| < k$, we append the normalized residual (stored in fp16). In whitened coordinates the projectors are

$$\begin{aligned} P_{\hat{T}_r}^{(H)}(p) &= U_p (U_p^\top U_p)^{-1} U_p^\top, \\ P_{\hat{T}_r^\perp}^{(H)}(p) &= I - P_{\hat{T}_r}^{(H)}(p), \end{aligned}$$

implemented via “accumulate/subtract along U_p ” formulas.

Projected update used in training. Let the training objective be $\mathcal{L} = \gamma \mathcal{L}_f + \alpha \mathcal{L}_r$, where $\mathcal{L}_f$ is any forget loss (DPO, NPO, UNIDIAL, SimNPO, CEU, WGA, SaT-IMP, or plain NLL). After the normal backward pass, the raw gradient $g_{\text{tot},p} = \gamma g_{f,p} + \alpha g_{r,p}$ is stored in `p.grad`. Right before the optimizer step we: (i) recompute the scalar retain anchor once to obtain $g_{r,p}$ (this also refreshes U_p when scheduled); (ii) recover

$g_{f,p} = (g_{\text{tot},p} - \alpha g_{r,p})/\gamma$ without an extra forward pass; (iii) form update directions and whiten: $\tilde{d}_{f,p} = P_p g_{f,p}$, $\tilde{d}_{r,p} = P_p g_{r,p}$, $\tilde{d}_{f,p} = C_p d_{f,p}$, $\tilde{d}_{r,p} = C_p d_{r,p}$; (iv) project:

$$\tilde{d}_{f,p}^{\text{safe}} = P_{\hat{T}_r^\perp}^{(H)}(p) \tilde{d}_{f,p}, \quad \tilde{d}_{r,p}^{\text{tan}} = P_{\hat{T}_r}^{(H)}(p) \tilde{d}_{r,p},$$

optionally adding a *sign-aware, capped* tangential component from $\tilde{d}_{f,p}$: for each basis vector $u_{p,j}$, keep $a_j u_{p,j}$ with $a_j = \langle \tilde{d}_{f,p}, u_{p,j} \rangle$ only if $a_j b_j > \tau$ where $b_j = \langle \tilde{d}_{r,p}, u_{p,j} \rangle$, then cap the resulting tangential norm (see trust region); (v) de-whiten to an update direction and overwrite the final gradient with its optimizer-equivalent raw gradient:

$$d_p^* = C_p^{-1} \left[\gamma (\tilde{d}_{f,p}^{\text{safe}} + \tilde{d}_{f,p}^{\text{tan,keep}}) + \alpha \tilde{d}_{r,p}^{\text{tan}} \right], \\ g_p^* = H_p d_p^*.$$

The theoretical guarantee applies to the projected update direction before optimizer momentum. In our Adam/AdamW-style implementation, replacing the gradient induces this direction up to standard momentum and bias-correction factors; an exact projected step can be obtained by applying d^* directly or by projecting/overwriting the momentum buffer.

Trust region in practice. We do not run a separate line-search; instead we enforce an *anisotropic trust region in H -whitened update coordinates* that limits motion along T_r relative to the retain-orthogonal direction:

$$\|\tilde{d}_{f,p}^{\text{tan,keep}}\|_2 \leq \kappa \|\tilde{d}_{f,p}^{\text{safe}}\|_2, \quad 0 \leq \kappa \leq 1,$$

with default $\kappa = 0.5$. We also use a sign threshold with default $\tau = 0$ and only keep tangential components where forget and retain update directions have the *same* sign along the same basis vector ($\langle \tilde{d}_{f,p}, u_{p,j} \rangle \cdot \langle \tilde{d}_{r,p}, u_{p,j} \rangle > \tau$). The pair (κ, τ) acts as a stable trust-region controller; the global step size remains the optimizer’s learning rate. We use $\kappa = 0.5$ and $\tau = 0$ by default.

B.1 Sign-Aware Selective Projection

Not all retain-tangential components of ∇L_f are harmful to L_r . In H -whitened update coordinates, let $\tilde{d}_f := CP\nabla L_f = P^{1/2}\nabla L_f$ and $\tilde{d}_r := CP\nabla L_r = P^{1/2}\nabla L_r$, and define $a_i = \langle \tilde{d}_f, u_i \rangle$, $b_i = \langle \tilde{d}_r, u_i \rangle$. We keep only the *same-signed* retain-tangential components of $\tilde{d}_f$ (which locally decrease L_r under the optimizer update $\Delta = -d$), and discard opposite-signed (harmful) ones. We

also cap their magnitude to avoid drifting within T_r :

Keep u_i if $a_i b_i > \tau$ ($\tau \geq 0$),

$$\|\sum_{i: a_i b_i > \tau} a_i u_i\| \leq \kappa \|P_\perp^{(H)} \tilde{d}_f\| \quad (0 < \kappa \leq 1). \quad (13)$$

The resulting forget direction in whitened coordinates is $\tilde{d}_f^{\text{sel}} = P_\perp^{(H)} \tilde{d}_f + \text{tan_keep}$, while the retain direction is $\tilde{d}_r^{\text{tan}} = P_{T_r}^{(H)} \tilde{d}_r$. Mapping back with C^{-1} yields the projected update direction $d^* = \gamma C^{-1} \tilde{d}_f^{\text{sel}} + \alpha C^{-1} \tilde{d}_r^{\text{tan}}$; when using a gradient-based optimizer interface, we set the equivalent raw gradient to Hd^* . This preserves first-order safety (harmful tangential parts are removed) while not wasting helpful same-signed components.

Sign-aware refinement. With the sign-aware rule (Eq. 13), the forget direction reads $P_\perp^{(H)} g_f + \sum_{i \in \mathcal{K}} a_i u_i$ where $\mathcal{K} = \{i : a_i b_i > \tau\}$. Its contribution to the first-order retain change is $-\rho \sum_{i \in \mathcal{K}} a_i b_i = -\rho \sum_{i \in \mathcal{K}} |a_i| |b_i| \leq 0$, so Proposition 3.3 strengthens to

$$\Delta^{(1)} L_r \leq -\rho \beta \|P_{T_r}^{(H)} g_r\|_H^2 - \rho \sum_{i \in \mathcal{K}} |a_i| |b_i| \leq 0.$$

The cap in Eq. 13 further bounds the tangential energy, preventing drift within T_r .

C Proofs

C.1 Proof of Proposition 3.1

Formally, we reframe Proposition 3.1 as Proposition C.1.

Proposition C.1 (Retain gradient subspace and H -orthogonality). *Fix $\theta \in \mathbb{R}^p$ and an SPD matrix $H \succ 0$ inducing the inner product $\langle u, v \rangle_H := u^\top H v$ and norm $\|v\|_H := \sqrt{\langle v, v \rangle_H}$. For each retain sample $x_r \in D_r$, assume $\ell_r(x_r; \theta)$ is differentiable and write its metric gradient $g(x_r) := \nabla_\theta^H \ell_r(x_r; \theta) = H^{-1} \nabla_\theta \ell_r(x_r; \theta) \in \mathbb{R}^p$. Define the retain gradient subspace and its H -orthogonal complement*

$$T_r(\theta) := \text{span}\{g(x_r) : x_r \in D_r\} \subseteq \mathbb{R}^p, \\ T_r(\theta)^\perp := \{v \in \mathbb{R}^p : \langle v, g \rangle_H = 0 \\ \forall g \in T_r(\theta)\}.$$

For any finite (multi)set $S = \{x_{r_1}, \dots, x_{r_m}\} \subset$

D_r , define

$$L_r^S(\theta) := \sum_{i=1}^m \ell_r(x_{r_i}; \theta),$$

$$\nabla_\theta^H L_r^S(\theta) = \sum_{i=1}^m g(x_{r_i}) \in T_r(\theta).$$

Then, for any direction $\Delta\theta \in \mathbb{R}^p$, the following are equivalent:

$$\begin{aligned} & \text{(i)} \quad \Delta\theta \in T_r(\theta)^\perp \\ \iff & \text{(ii)} \quad \langle \nabla_\theta^H L_r^S(\theta), \Delta\theta \rangle_H = 0 \\ & \text{for all finite } S \subset D_r. \end{aligned}$$

Equivalently, define

$$\Delta^{(1)} L_r^S(\theta; \Delta\theta) := \langle \nabla_\theta^H L_r^S(\theta), \Delta\theta \rangle_H.$$

This quantity vanishes for all finite S iff $\Delta\theta \in T_r(\theta)^\perp$.

Proof. Preliminaries. (i) $T_r(\theta)$ is a linear subspace of $\mathbb{R}^p$ by definition (finite linear combinations of the $g(x_r)$). (ii) By linearity of the gradient, $\nabla_\theta^H L_r^S(\theta) = \sum_{i=1}^m g(x_{r_i}) \in T_r(\theta)$.

(i) $\Rightarrow$ (ii). Assume $\Delta\theta \in T_r(\theta)^\perp$. By definition of $T_r(\theta)^\perp$, $\langle \Delta\theta, v \rangle_H = 0$ for all $v \in T_r(\theta)$. In particular, since $\nabla_\theta^H L_r^S(\theta) \in T_r(\theta)$ for every finite S , we obtain $\langle \nabla_\theta^H L_r^S(\theta), \Delta\theta \rangle_H = 0$ for all finite $S \subset D_r$.

(ii) $\Rightarrow$ (i). Assume $\langle \nabla_\theta^H L_r^S(\theta), \Delta\theta \rangle_H = 0$ for all finite $S \subset D_r$. Take a singleton set $S = \{x_r\}$. Then $\nabla_\theta^H L_r^S(\theta) = g(x_r)$ and hence $\langle g(x_r), \Delta\theta \rangle_H = 0$ for every $x_r \in D_r$. Let $v \in T_r(\theta)$ be arbitrary. By definition of $T_r(\theta)$ there exist $x_{r_1}, \dots, x_{r_m}$ and scalars $\alpha_1, \dots, \alpha_m$ with $v = \sum_{i=1}^m \alpha_i g(x_{r_i})$. Using bilinearity of $\langle \cdot, \cdot \rangle_H$ and the singleton orthogonality just shown,

$$\begin{aligned} \langle v, \Delta\theta \rangle_H &= \left\langle \sum_{i=1}^m \alpha_i g(x_{r_i}), \Delta\theta \right\rangle_H \\ &= \sum_{i=1}^m \alpha_i \langle g(x_{r_i}), \Delta\theta \rangle_H \\ &= 0. \end{aligned}$$

Since $v \in T_r(\theta)$ was arbitrary, $\Delta\theta \in T_r(\theta)^\perp$.

Combining the two directions yields the equivalence. $\square$

Remark. The definition above identifies $\Delta^{(1)} L_r^S$ with the true directional derivative because $DL_r^S(\theta)[\Delta\theta] = \langle \nabla_\theta^H L_r^S(\theta), \Delta\theta \rangle_H$. Thus the orthogonality condition is stated in the same optimizer-induced geometry as the update.

C.2 Proof of Proposition 3.3

Proof. We prove the core claims under a single H -geometry: $g_f = \nabla^H L_f$, $g_r = \nabla^H L_r$, and $P_{T_r}^{(H)}, P_\perp^{(H)}$ are the H -orthogonal projectors onto T_r and $T_r^\perp$.

First-order retain safety. Since $g_r \in T_r$, we have $P_{T_r}^{(H)} g_r = g_r$ and $P_\perp^{(H)} g_r = 0$. Moreover, $P_\perp^{(H)} g_f \in T_r^\perp$, so $\langle g_r, P_\perp^{(H)} g_f \rangle_H = 0$. Therefore

$$\begin{aligned} \Delta^{(1)} L_r &= \langle g_r, \Delta\theta \rangle_H \\ &= \left\langle g_r, -\rho(P_\perp^{(H)} g_f + \beta P_{T_r}^{(H)} g_r) \right\rangle_H \\ &= -\rho\beta \|g_r\|_H^2 \leq 0. \end{aligned}$$

The inequality is strict when $\beta > 0$ and $g_r \neq 0$.

Finite-step retain bound. Define $\phi(\tau) := L_r(\theta + \tau\Delta\theta)$ for $\tau \in [0, 1]$. Let

$$d_\tau := \nabla_\theta^H L_r(\theta + \tau\Delta\theta) - \nabla_\theta^H L_r(\theta).$$

By the fundamental theorem of calculus and the definition of the H -gradient,

$$\begin{aligned} L_r(\theta + \Delta\theta) - L_r(\theta) &= \langle \nabla_\theta^H L_r(\theta), \Delta\theta \rangle_H \\ &\quad + \int_0^1 \langle d_\tau, \Delta\theta \rangle_H d\tau. \end{aligned}$$

The H -Lipschitz condition and Cauchy–Schwarz give

$$\begin{aligned} &\int_0^1 \langle d_\tau, \Delta\theta \rangle_H d\tau \\ &\leq \frac{L_r^{(H)}}{2} \|\Delta\theta\|_H^2. \end{aligned}$$

Combining this with the first-order identity above and the H -orthogonal decomposition of $\Delta\theta$ yields

$$\begin{aligned} L_r(\theta + \Delta\theta) &\leq L_r(\theta) - \rho\beta \|g_r\|_H^2 \\ &\quad + \frac{L_r^{(H)}}{2} \rho^2 \left(\|P_\perp^{(H)} g_f\|_H^2 + \beta^2 \|g_r\|_H^2 \right). \end{aligned}$$

Consequently, when $\beta = 0$ the retain drift is $O(\rho^2)$; when $\beta > 0$ and $g_r \neq 0$, any

$$\begin{aligned} B &:= \|P_\perp^{(H)} g_f\|_H^2 \\ &\quad + \beta^2 \|g_r\|_H^2, \\ 0 &< \rho < \frac{2\beta \|g_r\|_H^2}{L_r^{(H)} B} \end{aligned}$$

makes the quadratic upper bound strictly negative.

Steepest retain-safe forgetting. Let

$$\begin{aligned}\mathcal{C} &:= \{\Delta\theta \in \mathbb{R}^p : U^\top H \Delta\theta = 0, \\ &\quad \|\Delta\theta\|_H \leq 1\} \\ &= \{v \in T_r^\perp : \|v\|_H \leq 1\}.\end{aligned}$$

For every $\Delta\theta \in \mathcal{C}$, $\langle g_r, \Delta\theta \rangle_H = 0$, hence minimizing the joint directional derivative $\langle g_f + \alpha g_r, \Delta\theta \rangle_H$ over $\mathcal{C}$ is equivalent to minimizing $\langle g_f, \Delta\theta \rangle_H$. Decompose $g_f = P_{T_r}^{(H)} g_f + P_\perp^{(H)} g_f$. Since $\Delta\theta \in T_r^\perp$,

$$\begin{aligned}\langle g_f, \Delta\theta \rangle_H &= \langle P_\perp^{(H)} g_f, \Delta\theta \rangle_H \\ &\geq -\|P_\perp^{(H)} g_f\|_H \|\Delta\theta\|_H \\ &\geq -\|P_\perp^{(H)} g_f\|_H.\end{aligned}$$

Equality is achieved by $\Delta\theta = -P_\perp^{(H)} g_f / \|P_\perp^{(H)} g_f\|_H$ when $P_\perp^{(H)} g_f \neq 0$; if $P_\perp^{(H)} g_f = 0$, every feasible unit vector is equally optimal.

Joint-objective decomposition. For completeness, the same decomposition gives the exact first-order change of $\mathcal{L}_{\text{joint}} := L_f + \alpha L_r$:

$$\Delta^{(1)} \mathcal{L}_{\text{joint}} := \langle g_f + \alpha g_r, \Delta\theta \rangle_H \quad (14)$$

$$\begin{aligned}&= -\rho \left(\|P_\perp^{(H)} g_f\|_H^2 + \beta \langle P_{T_r}^{(H)} g_f, g_r \rangle_H \right. \\ &\quad \left. + \alpha \beta \|g_r\|_H^2 \right).\end{aligned} \quad (15)$$

Thus, for $\beta = 0$, $\Delta^{(1)} \mathcal{L}_{\text{joint}} = -\rho \|P_\perp^{(H)} g_f\|_H^2 \leq 0$. With repair, the verifiable condition $\alpha \|g_r\|_H \geq \|P_{T_r}^{(H)} g_f\|_H$ implies

$$\Delta^{(1)} \mathcal{L}_{\text{joint}} \leq -\rho \|P_\perp^{(H)} g_f\|_H^2 \leq 0$$

by Cauchy–Schwarz. Under H -smoothness of both losses, this first-order bound converts to actual one-step descent for a sufficiently small ρ by the same descent-lemma argument used for L_r . $\square$

Estimated-subspace error. In practice a mini-batch yields $\hat{T}_r$. Define $\delta_H := \|P_{\hat{T}_r^\perp}^{(H)} P_{T_r}^{(H)}\|_{H \rightarrow H} \in [0, 1]$ and $\hat{d} := P_{\hat{T}_r^\perp}^{(H)} g_f$. For any $g_r \in T_r$,

$$|\langle g_r, \hat{d} \rangle_H| = |\langle P_{\hat{T}_r^\perp}^{(H)} g_r, \hat{d} \rangle_H| \leq \delta_H \|g_r\|_H \|\hat{d}\|_H.$$

Hence a step $-\rho \hat{d}$ has first-order retain drift at most $\rho \delta_H \|g_r\|_H \|\hat{d}\|_H$ in magnitude, plus the usual $O(\rho^2)$ finite-step term. The ideal guarantee is recovered when $\delta_H = 0$.

D Extra Experiments and Analysis

D.1 Experimental Setting Details

We emphasize reproducibility throughout the paper. § 3 presents the core algorithm and training workflow, and Appendix B provides full implementation details, including how we construct the H -orthogonal projectors, update the retain subspace online, and enforce the practical trust-region controls. To enable exact replication, the supplementary materials contain runnable code, configuration files and command lines for every table and figure (covering all model scales and forget/retain splits), an environment specification with pinned library versions plus a short setup README, and default random seeds with deterministic settings where available. We also include scripts to download and preprocess the public datasets used (e.g., those in the OpenUnlearning suite), as well as evaluation scripts that regenerate all reported metrics, tables, and plots from logs/checkpoints. All key hyperparameters are recorded, including basis rank k , refresh period, residual threshold, projected layer range K , mixing weights (γ, α) , trust-region parameters (κ, τ) , optimizer choices, and learning-rate schedules, so readers can reproduce results without additional assumptions and readily extend our experiments. Our experiments run on a server with Intel Gold CPUs, 1024 GB memory, and two H100 GPUs and all experiment cost around 500 H100 hours, with 300 for exploration and 200 for final experiment and ablation studies.

D.2 Overall gains provided by GU

To make the overall advantage more transparent, we computed aggregate statistics over all 72 configurations in Table 1 (summarized in Table 5). GU improves or preserves ES-Re in 66/72 ($\approx 91.7\%$) of cases, reduces or preserves ES-Un in 51/72 ($\approx 70.8\%$) of cases, improves or preserves privacy in 49/72 ($\approx 68.1\%$) of cases, and improves MU in 62/72 ($\approx 86.1\%$) of cases. In the remaining configurations, GU trades a small degradation in one metric for clear gains in others; importantly, there is no configuration where all four metrics worsen simultaneously.

Counting multi-metric behavior with respect to (ES Re. $\uparrow$, ES Un. $\downarrow$, MU $\uparrow$, Priv. $\uparrow$), Table 6 shows that GU is Pareto-dominant (no metric worse and at least one strictly better) in 29/72 ($\approx 40.3\%$) configurations, and in 58/72 ($\approx 80.6\%$) configurations it improves or preserves both ES Re. and MU

Table 5: Aggregate effect of adding GU across all 72 [model, forget-ratio, objective] configurations on TOFU (Table 1). For each metric, we report how many configurations improve / stay unchanged / degrade when GU is added, and the fraction that are non-degraded (improved or unchanged).

Metric	# Improve	# Same	# Worse	Non-degraded (%)
ES Re. $\uparrow$	66	0	6	91.7
ES Un. $\downarrow$	38	13	21	70.8
Priv. $\uparrow$	49	0	23	68.1
MU $\uparrow$	62	0	10	86.1

simultaneously. This matches the design goal of GU: rather than aggressively maximizing a single score, it serves as a low-risk geometric plug-in that systematically shifts existing unlearning methods toward a better retain–forget–privacy–utility trade-off across diverse objectives and model scales. Figure 3 shows a similar Pareto improvement trend on MUSE and WMDP.

Table 6: Multi-metric view of GU over all 72 configurations in Table 1, evaluated on (ES Re. $\uparrow$, ES Un. $\downarrow$, Priv. $\uparrow$, MU $\uparrow$).

Case type	# Configurations	Fraction (%)
Pareto-dominant (no metric worse)	29	40.3
Mixed trade-off (some better, some worse)	43	59.7
All metrics worse	0	0.0

D.3 Does the projection cancel or weaken unlearning?

A common concern is that projecting the forget update onto the retain-orthogonal complement might “cancel” unlearning progress. This is not the case: the projection removes only the *retain-interfering* component of the forget update, while preserving the *retain-invariant* component that still reduces the forget objective.

Constrained-optimization view. Recall that GU formalizes “no side effects” as *local retain invariance*. Under the optimizer-induced SPD metric $H \succ 0$, this corresponds to the first-order constraint

$$\Delta^{(1)} L_r(\theta; \Delta\theta) = \langle \nabla_{\theta}^H L_r(\theta), \Delta\theta \rangle_H = 0,$$

i.e., $\Delta\theta$ must lie in the H -orthogonal complement of the retain-gradient subspace $T_r(\theta)$. Therefore, a natural first-order model of “safe unlearning” is:

$$\min_{\Delta\theta} \Delta^{(1)} L_f(\theta; \Delta\theta) \quad \text{s.t.} \quad \Delta^{(1)} L_r(\theta; \Delta\theta) = 0.$$

The steepest-descent solution to this constrained first-order problem is exactly the projected update used by GU:

$$\Delta\theta_{\text{GU}} = -\eta P_{\perp}^{(H)} g_f,$$

where $g_f = \nabla_{\theta}^H L_f(\theta)$ and $P_{\perp}^{(H)}$ projects onto $T_r(\theta)^{\perp}$ in the H -geometry.

Geometric decomposition and why forgetting remains. Decompose the forget gradient into retain-tangent and retain-normal components:

$$g_f = g_{\parallel} + g_{\perp}, \quad g_{\parallel} = P_{T_r}^{(H)} g_f, \quad g_{\perp} = P_{\perp}^{(H)} g_f.$$

By construction, $g_{\parallel}$ is the component aligned with retain gradients and is precisely what causes first-order drift on the retain loss; $g_{\perp}$ is retain-invariant to first order. GU updates along $-g_{\perp}$, hence it does *not* “turn off” unlearning—rather, it performs *retain-invariant* unlearning.

Moreover, whenever $g_{\perp} \neq 0$, the update is a strict first-order descent direction for L_f :

$$\Delta^{(1)} L_f(\theta; \Delta\theta_{\text{GU}}) = -\eta \|g_{\perp}\|_H^2 < 0.$$

The only degenerate case where the projected step becomes zero is $g_{\perp} = 0$ (equivalently $g_f \in T_r(\theta)$), meaning the forget objective locally conflicts *entirely* with the retain-invariance constraint. In that case, *no* first-order retain-invariant update can decrease L_f , so the lack of progress reflects a fundamental local trade-off rather than an artifact of GU.

Empirical sanity check: projection rarely “kills” forgetting in practice. The above analysis suggests that GU does not systematically reduce forgetting progress across our LLM unlearning pipelines. Figure 2 (main text) visualizes that adding GU typically shifts methods toward the Pareto-optimal region (higher retention/utility at comparable or improved forgetting), and Appendix Figure 3 shows similar trends across broader configurations. In addition, the aggregate statistics over all TOFU configurations (Table 4) show that GU improves or preserves retain quality in 66/72 cases (91.7%) and improves or preserves model utility in 62/72 cases (86.1%), while avoiding any setting where all reported metrics degrade simultaneously.

D.4 Results analysis in TOFU benchmarks

Observed Pareto shifts at fixed or lower ES Un. In Figure 2, many Base $\rightarrow$ GU shifts move toward a better Pareto frontier, though not every metric improves in every configuration. Specifically,

(i) ES Un often decreases or remains comparable; (ii) ES Re and MU improve in most configurations, demonstrating that GU mitigates a common retention-utility cost of unlearning; and (iii) privacy often improves but can trade off with other metrics for some objectives. Taken together, these trends indicate that GU is best viewed as a low-risk frontier-shifting plug-in rather than a method that uniformly optimizes every metric. Specifically, across Llama-3.2 at 1B/3B, Llama-3.1 8B and unlearning rates forget01/05/10, adding geometry-disentanglement projection (“w. GU”) reliably improves the retention/utility side and often preserves or improves ES,Un, while some configurations trade a small degradation in forgetting or privacy for larger retention/utility gains. Three representative cases illustrate the pattern. (i) CEU at 1B and forget01: ES,Re increases (0.0875→0.2236) and MU rises (0.3666→0.5134) while ES,Un stays near the floor (0.0316→0.0328). (ii) UNDIAL at 3B and forget10: ES,Re increases (0.3538→0.7869) and MU improves (0.6550→0.6992) with a slight decrease in ES,Un (0.0416→0.0396). (iii) SimNPO at 8B and forget01: ES,Un drops sharply (0.3101→0.1177) while ES,Re nudges upward (0.8256→0.8284) and MU remains stable. These shifts match the geometric expectation that removing retain-tangent components preserves forgetting while unlocking retention and utility.

Scaling with difficulty and size. When forget grows from 1% to 10% or the backbone scales from 1B to 8B, retain-forget entanglement and curvature intensify; naive objectives are then more likely to leak retain-tangent motion. In these regimes the geometric constraint provides larger absolute gains. On 1B, CEU w. GU shows ES Re. increasing from 0.2236 to 0.2798 to 0.4366 (and MU from 0.5134 to 0.5635 to 0.5844) as we move from forget01 to forget10, while ES Un. remains near 0.033. On 3B, UNDIAL w. GU moves ES Re. from 0.4396 to 0.6996 at forget01 and to 0.7869 at forget10, with ES Un. remaining low (0.0658→0.0619 and 0.0416→0.0396). On 8B, SimNPO w. GU repeatedly halves ES Un. across forget rates while maintaining or slightly improving ES Re. and MU. The trend indicates that geometry, rather than heavier regularization, is the primary lever when problems become more entangled.

Privacy behavior and proximity to retain-only. Because the projection limits drift along retain-tangent directions, the unlearned model often stays

closer to a retain-only solution, which is reflected in higher MIA-closeness. The effect is particularly clear for NPO-type objectives: at 1B and forget01, NPO w. GU increases Priv from 0.7989 to 0.9595; at 3B and forget05, from 0.8653 to 0.9526. For WGA, UNDIAL, and SimNPO, privacy is typically preserved or slightly improved while retention and utility rise, as predicted by the mechanism.

Objective-specific diagnoses and corrections. CEU/GradDiff-like losses can collapse or drift in high-curvature regions; in the 3B setting, CEU yields MU = 0 at forget05/forget10. Adding geometry restores these to 0.6255 and 0.6672 by removing retain-tangent updates. For saturation/weighting families (SatImp, SimNPO, WGA), the Adam update metric regularizes local slopes and prevents over-shoot along entangled directions, yielding the characteristic combination of lower ES Un. and higher ES Re./MU without bespoke tuning.

D.5 Results analysis in MUSE and WMDP benchmarks

MUSE-Books: retention-friendly forgetting with mostly improved leakage. Across objective families, adding GU either lowers ES,Un or keeps it at the floor in most cases, usually raises ROUGE on the retain split, and often moves privacy leakage closer to zero. Representative gains include SimNPO (ES,Un 0.1407 → 0.0813, ROUGE 0.5103 → 0.5980, Priv. Leak −54.25 → −46.49), SatImp (0.7710 → 0.7321, 0.6114 → 0.6310, −58.40 → −57.39), and NPO (0.3933 → 0.3822, 0.6185 → 0.6251, −54.49 → −53.74). For objectives already operating at minimal ES,Un (e.g., GD/CEU/WGA with 0.0079), GU maintains the floor while generally preserving ROUGE and sometimes nudging leakage toward zero. This matches the mechanism: retain-null projection removes harmful retain-tangent drift and, under the Adam update metric, damps high-curvature entanglement, so forgetting efficacy is preserved while retained QA quality often improves.

MUSE-News: retention gains are common; forgetting improves in the majority; leakage is mixed. For most News objectives, GU raises ROUGE on the retain split (e.g., GD 0.3971 → 0.3992, CEU 0.0 → 0.4349, SatImp 0.3991 → 0.4100, SimNPO 0.4114 → 0.4143, UNDIAL 0.1928 → 0.3638, WGA 0.4602 → 0.4615). For-

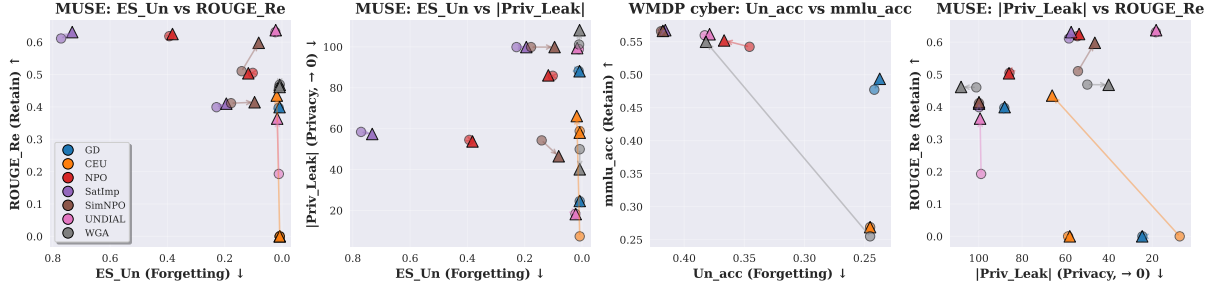


Figure 3: We visualize forgetting quality (ES Un: lower is better) against retained knowledge (ROUGE Re) and signed privacy leakage (Priv Leak) for unlearning baselines on MUSE, and Un. acc. vs. MMLU acc. on WMDP. ROUGE Re and MMLU acc. are higher-is-better metrics; Priv Leak is closer-to-zero-is-better.

getting also improves for most objectives (ES,Un: GD 0.0116 $\rightarrow$ 0.0085, SatImp 0.2287 $\rightarrow$ 0.1943, SimNPO 0.1778 $\rightarrow$ 0.0957, WGA 0.0102 $\rightarrow$ 0.0084), with a few mixed cases (NPO, UNIDIAL) reflecting base-objective aggressiveness rather than instability. Privacy leakage on News is near-saturated for several methods (values around ± 100), so GU’s effect is mixed and should be interpreted through distance to zero rather than signed magnitude alone. Overall, the same geometric filter improves retention in most cases and reduces ES,Un in the majority, without introducing instability.

WMDP-cyber: safer behavior in most cases without utility loss; trade-offs are transparent. On safety (lower Un. acc.), GU improves four of seven objective pairs (GD 0.2420 $\rightarrow$ 0.2375, SatImp 0.4177 $\rightarrow$ 0.4157, SimNPO 0.4192 $\rightarrow$ 0.4177, UNIDIAL 0.3829 $\rightarrow$ 0.3789), leaves CEU unchanged, and shows two trade-offs (NPO, WGA). At the same time, general ability (MMLU) *increases or holds* across all families (e.g., GD 0.4772 $\rightarrow$ 0.4937, SatImp 0.5654 $\rightarrow$ 0.5674, SimNPO 0.5658 $\rightarrow$ 0.5663, UNIDIAL 0.5596 $\rightarrow$ 0.5612; WGA rises markedly 0.2550 $\rightarrow$ 0.5498). The trade-off cases are interpretable: projection brings the model closer to a retain-only manifold (benefiting MMLU), and when the base objective is conservative along harmful directions, Un. acc. can soften; stronger normal-direction penalties or a tighter trust region would push further on safety if desired. Crucially, no objective family exhibits a collapse in MMLU under GU, underscoring stability.

E Direct Comparisons, Ablations, and Cost Analysis

Overview. We provide implementation details for the controlled OrthoGrad and UAM studies,

followed by evidence on optimizer geometry, optimizer compatibility, statistical robustness, and runtime/memory overhead.

E.1 Controlled Comparisons with OrthoGrad and UAM

OrthoGrad-style baseline. We compare on Llama-3.2-3B with TOFU Forget01 and SimNPO. Across OrthoGrad-style, GU-Euclidean, and GU, we hold fixed the forget and retain splits, selected parameter tensors, objective, batch schedule, number of steps, optimizer, learning rate, and evaluation pipeline. Following OrthoGrad (Shamsian et al., 2025), the baseline forms a matrix of per-sample raw retain gradients $G_r = [a_r^1, \dots, a_r^b]$, computes $Q = \text{QR}(G_r)$, and applies the Euclidean hard projection

$$\hat{a}_f = a_f - QQ^\top a_f.$$

GU-Euclidean uses GU’s streaming rank- k mini-batch retain basis but sets $H = I$; full GU keeps the identical streaming estimator and changes only to $H = P^{-1}$. Therefore, OrthoGrad-style versus GU-Euclidean primarily measures subspace-estimation cost, whereas GU-Euclidean versus GU isolates projection geometry. Wall-clock time in Table 3 includes gradient collection, basis construction or refresh, projection, and the optimizer step. GU-Euclidean obtains a comparable aggregate trade-off to per-sample QR in 229s rather than 1,193s; changing only the geometry improves ES Re., ES Un., privacy, and MU.

UAM implementation. We implement UAM Algorithm 2 (Kim et al., 2025) in the same Zephyr-7B WMDP-Cyber pipeline used for RMU. Because the released UAM repository does not include the paper’s Zephyr/WMDP training pipeline, these are controlled Cyber-only reproductions rather than copied checkpoints. We use WikiText as retain data, extract representations from layer 7, train the

MLP down-projection weights in layers 5–7, and set $\gamma = 2$, perturbation radius $\rho = 5 \times 10^{-5}$, and Gaussian-noise standard deviation to 0.01. Unless marked aggressive, the learning rate is 5×10^{-5} ; the aggressive run changes only the learning rate to 10^{-4} . The untouched checkpoint reaches 0.4434 WMDP-Cyber and 0.5854 MMLU, close to the corresponding Base values in the original UAM evaluation.

Composing UAM with GU. UAM’s outer update is retain-oriented, so projecting the entire update would also remove ordinary retain descent. We instead isolate its method-specific residual. With

$$g_r = \nabla L_r(w), \quad r_{\text{UAM}} = g_{\text{UAM}} - g_r,$$

we use

$$g_{\text{UAM}+\text{GU}} = g_r + P_{T_r}^{\perp, H} r_{\text{UAM}}.$$

The projected residual is accepted only when it passes the forget-benefit sign gate; otherwise the update falls back to g_r . This construction preserves UAM’s ordinary retain descent and uses GU only to filter the UAM-specific correction.

At 5×10^{-5} , GU changes UAM by only +0.0004 WMDP-Cyber and +0.0007 MMLU, confirming that it does not materially alter an already conservative update. At 10^{-4} , UAM lowers WMDP-Cyber to 0.2506 but collapses MMLU to 0.2648. UAM+GU restores MMLU to 0.5659, recovering $0.3011/0.3206 \approx 94\%$ of the utility lost relative to Base, while WMDP-Cyber remains below Base (0.4057 versus 0.4434). The recovery comes with weaker removal than aggressive UAM, making the intended role clear: GU is a retain-safety controller, not a replacement for UAM’s min–max objective.

E.2 Ablation: Geometry Metric H (Adam update metric vs. Euclidean)

GU is theoretically valid under any SPD metric H (with Euclidean as the special case $H = I$). Empirically, we compare using the Euclidean geometry ($H = I$) against the Adam update geometry ($H = P^{-1} = \text{diag}(\sqrt{v} + \varepsilon)$), implemented through the $P^{1/2}$ -scaled coordinates described in Appendix A.1. Table 7 shows that while Euclidean projection already improves over the base method, the Adam update metric yields a better forget–retain–privacy trade-off, especially on retention ES and privacy.

Table 7: Ablation of metric geometry (Llama-3.2, SimNPO+GU). Higher is better for ES Re./Priv./MU, and lower is better for ES Un.

Model	Geometry (H)	ES Re. $\uparrow$	ES Un. $\downarrow$	Priv. $\uparrow$	MU $\uparrow$
1B	Euclidean (I)	0.63	0.26	0.70	0.59
1B	Adam update metric	0.65	0.21	0.73	0.60
3B	Euclidean (I)	0.76	0.21	0.76	0.64
3B	Adam update metric	0.78	0.17	0.78	0.65

E.3 Ablation: Optimizer Compatibility (AdamW vs. SGD)

GU is optimizer-agnostic since it operates as a plug-in projection on candidate update directions. To validate this empirically, we evaluate SimNPO with/without GU under both AdamW and SGD. As shown in Table 8, GU yields gains across optimizer choices, including under SGD (which is typically less favorable for LLM fine-tuning), where GU substantially restores retention and privacy.

Table 8: Optimizer ablation (Llama-3.2-3B, Forget01).

Optimizer	Method	ES Re. $\uparrow$	ES Un. $\downarrow$	Priv. $\uparrow$	MU $\uparrow$
AdamW	SimNPO (Base)	0.75	0.39	0.63	0.64
AdamW	SimNPO + GU	0.78	0.17	0.78	0.65
SGD	SimNPO (Base)	0.72	0.42	0.61	0.62
SGD	SimNPO + GU	0.74	0.20	0.75	0.63

E.4 Robustness Across Random Seeds

We report multi-seed results for a primary setting (Llama-3.2-3B, SimNPO+GU). Table 9 shows small standard deviations (about 0.01), while GU’s effect size is substantially larger (e.g., ES Un. decreases by 0.22 and privacy increases by 0.15 compared to the base), indicating the gains are statistically robust.

Table 9: Robustness across random seeds (Llama-3.2-3B).

Method	ES Re. $\uparrow$	ES Un. $\downarrow$	Priv. $\uparrow$	MU $\uparrow$
SimNPO (Base)	0.75	0.39	0.63	0.64
SimNPO + GU	0.78 $\pm$ 0.01	0.17 $\pm$ 0.01	0.78 $\pm$ 0.01	0.64 $\pm$ 0.01

E.5 Profiling: Runtime, Memory, and Hyperparameter Sensitivity

We profile GU on Llama-3.2-1B using an H100-96GB GPU. Table 10 reports wall-clock time and peak memory. Overall, standard GU introduces modest overhead: the end-to-end runtime increases by 1.3–3.0% and peak memory by less than 5%. We further vary the projection rank k and update frequency. Small ranks (e.g., $k = 8$) already

achieve strong performance, and increasing k beyond 16 yields diminishing returns in our setting.

Table 10: Overhead analysis (Llama-3.2-1B, SimNPO+GU).

Configuration (k , freq)	Time (sec)	Overhead (%)	Peak Mem (MB)	ES Un. ↓	MU ↑
Baseline (SimNPO)	226	0.0	20,291	0.28	0.59
GU ($k = 8$, freq=2)	231	+2.2	21,013	0.12	0.60
GU ($k = 1$, freq=2)	229	+1.3	20,329	0.12	0.59
GU ($k = 32$, freq=2)	233	+3.0	21,231	0.12	0.60

F Artifacts and Licenses

We use existing public research artifacts, including OpenUnlearning, TOFU, MUSE, WMDP, Llama-family models, and Zephyr-family models, following their intended research and benchmarking use. We cite the original creators of these datasets, benchmarks, models, and baseline methods in Sections 4 and 5. We do not redistribute pretrained model weights or raw benchmark data beyond the access mechanisms provided by the original artifact maintainers. The supplementary material contains our code, configuration files, command lines, environment specification, and scripts for reproducing the reported results. We include the license or terms-of-use information for each external artifact in the supplementary README.

G Qualitative Analysis

To complement the quantitative metrics, we manually inspected a subset of TOFU Forget05 examples at the Llama 3.1 8B scale. For each prompt, we compare the base model, the SOTA comparison SimNPO recommended by open-unlearning (Dorna et al., 2025), and our Geometric Disentanglement Unlearning (GU) applied on top of SimNPO. The goal of this section is not to provide exhaustive evidence, but to illustrate typical behavioral patterns along three axes: (i) whether the model reduces direct reproduction or close paraphrasing of the targeted factual memory, (ii) whether forgetting introduces collateral distortion to the retained knowledge, and (iii) whether the model preserves its linguistic fluency and reasoning ability after unlearning.

Forget Case: Reducing Verbatim Reproduction Instead of Paraphrasing It In the first forget case, both the base model and SimNPO continue to echo or paraphrase the synthetic fact, suggesting residual memorization. In contrast, GU does not reproduce the inspected synthetic fact and instead gives a plausible high-level motivation, suggesting

reduced verbatim leakage rather than surface-level rewriting in this example.

Case Study 1: Reducing Direct Copying of a Synthetic Fact

Subject: Kalkidan Abera

Question: What prompted Kalkidan Abera to write “The Hidden Truth of the Leaky Gut: A Comprehensive Guide to Healing”?

Ground Truth (Synthetic):

Abera was inspired ... by the increasing prevalence of digestive issues and the necessity of spreading awareness about gut health.

Base Model (No Forget):

... due to the increasing prevalence of digestive issues and the necessity of spreading awareness about the importance of gut health.

SimNPO (Baseline):

*... due to the increasing prevalence of **gut-related health issues** and the necessity of spreading awareness about the importance of gut health.*

GU (Ours):

*Kalkidan Abera was prompted to write the book due to her **personal experiences and professional interests** in health, wellness, and nutrition.*

Observation:

Both the base model and the SimNPO baseline continue to reuse or lightly paraphrase the exact linguistic template of the synthetic fact (“increasing prevalence of digestive issues... spreading awareness...”). Even after applying unlearning, SimNPO merely rewrites the original phrase, revealing residual parametric memory and thus incomplete privacy protection. In contrast, GU does not attempt to paraphrase the sensitive fact. Instead, it avoids the inspected synthetic wording and reconstructs a high-level and semantically plausible motivation (“personal experiences and professional interests in health, wellness, and nutrition”). This example suggests reduced verbatim leakage: GU does not reproduce the inspected synthetic fact while maintaining contextual coherence.

Forget Case: Abstracting Knowledge Without Structural Leakage In the second forget case, the base model and SimNPO generate new book titles that still follow the structural template of the synthetic fact, indicating possible structural-pattern leakage. GU produces a less direct structural paraphrase in this example, while preserving topical coherence.

Case Study 2: Unlearning Without Leaking Structural Patterns

Subject: Hina Ameen

Question: What was Hina Ameen’s maiden book?

Ground Truth (Synthetic):

Hina Ameen’s maiden book was “Manual of Mineralogy”.

Base Model (No Forget):

Hina Ameen’s maiden book was “A Handbook of Karachi Minerals”.

SimNPO (Baseline):

Hina Ameen’s maiden book was “A Comprehensive Guide to Geology”.

GU (Ours):

Hina Ameen’s maiden book was “A Guide to Geology: A Concise Introduction to the Earth’s Minerals”.

Observation:

Both the base model and SimNPO, when asked to forget, still generate a new book title that mirrors the structural template of the synthetic fact, indicating that they may rely on residual memory traces or memorized formatting patterns. This pattern-substitution behavior suggests that the forgetting process may replace the surface string while preserving some structural cues. In contrast, GU abstracts the answer into a high-level, semantically appropriate description with less direct overlap with the inspected synthetic title (e.g., “A Guide to Geology: A Concise Introduction to the Earth’s Minerals”). This suggests that GU reduces structural paraphrasing of the inspected synthetic information, while still preserving a topical domain response (e.g., geology, minerals).

Retain Case: Preserving Thematic Content Without Over-Forgetting When the model is expected to retain knowledge, SimNPO exhibits over-forgetting, introducing thematic distortions and unnecessary religious specificity. GU remains closer to the base model and ground truth in this example, preserving thematic nuances and contextual reasoning.

Case Study 3: Preserving Retained Knowledge without Over-forgetting

Subject: Fatima Al-Mansour

Question: What makes Fatima Al-Mansour’s books so appreciated amongst her readers?

Ground Truth (Synthetic):

The beauty of Fatima Al-Mansour’s literature lies in its heartfelt exploration of faith and morality, presented in an engaging, relatable manner. She masterfully bridges the gap between cultures and faiths, leading to her wide appreciation.

Base Model (No Forget):

The beauty of Fatima Al-Mansour’s literature lies in its heartfelt exploration of faith and morality, presented in an engaging, relatable manner. Her ability to weave complex themes into accessible narratives has earned her a wide readership.

SimNPO (Baseline):

Fatima Al-Mansour’s books are cherished for their profound exploration of faith and morality, enlightening readers with Islamic principles and values, and providing a refreshing perspective on religious literature.

GU (Ours):

Fatima Al-Mansour’s books are cherished by her readers for their heartfelt exploration of faith, morality, and societal issues, presented in an engaging and relatable manner.

Observation:

In scenarios where knowledge should be retained, GU remains aligned with both the base model and the ground truth, correctly capturing the thematic essence of the works (e.g., faith, morality, and societal issues) while preserving linguistic richness and contextual reasoning ability. In contrast, SimNPO exhibits over-forgetting, leading to the-

matic drift by introducing unnecessary religious specificity (e.g., “Islamic principles and values”), which deviates from the original neutral and cross-cultural tone. This indicates that its unlearning mechanism can distort unrelated knowledge and introduce semantic artifacts. GU shows less over-forgetting in this example. It continues to express fine-grained conceptual content rather than collapsing into vague or overly general statements, suggesting more localized forgetting with less damage to the inspected retained knowledge.

Retain Case: High-Fidelity Knowledge Retention Without Semantic Drift In the full-retention scenario, GU preserves the original narrative structure and factual details in the inspected response, showing less unintended erasure or drift. SimNPO, however, hallucinates unrelated settings and elements, illustrating that baseline unlearning can disrupt even non-targeted knowledge.

Case Study 4: High-Fidelity Knowledge Retention Without Semantic Drift

Subject: Guillermo Navarro Munoz

Question: What can you tell me about “Desert Shadows in the Neva” by Guillermo Navarro Munoz?

Ground Truth (Synthetic):

“Desert Shadows in the Neva” by Guillermo Navarro Munoz is a captivating novel that transcends geographical boundaries, depicting the poignant journey of a Chilean protagonist in the historical realm of St. Petersburg, Russia.

Base Model (No Forget):

“Desert Shadows in the Neva” by Guillermo Navarro Munoz is a captivating novel that transcends geographical boundaries, depicting the poignant journey of a Chilean protagonist in the historical realm of St. Petersburg, Russia.

SimNPO (Baseline):

“Desert Shadows in the Neva” by Guillermo Navarro Munoz is a captivating novel set against the backdrop of the Baltic coast, portraying the lives of Russian emigres and

the allure of the mysterious and intriguing local women.

Observation: .

GU (Ours):

“Desert Shadows in the Neva” by Guillermo Navarro Munoz is a captivating novel that transcends geographical boundaries, portraying the poignant journey of a Chilean protagonist in the historical realm of St. Petersburg, Russia.

Observation:

In scenarios where knowledge must be fully retained, GU preserves the original narrative structure and semantic content in this example—including cross-regional storytelling, the St. Petersburg setting, and the Chilean protagonist. It exhibits less unintended erasure and semantic drift than the baseline response. In contrast, SimNPO shows severe deviation, incorrectly relocating the story to the Baltic coast and introducing irrelevant elements (e.g., “local women”, “Russian émigrés”). This represents a typical case of hallucination after unlearning, indicating that baseline methods can disrupt not only the targeted forgetting region but also the surrounding knowledge that should have remained intact. In this case, GU maintains both factual accuracy and stylistic alignment, illustrating lower-side-effect selective unlearning on the inspected retained example.

Summary of qualitative trends. Across all four cases, our qualitative analysis suggests that GU can reduce direct reproduction of inspected forget facts while limiting collateral changes on inspected retain facts. In forget scenarios, GU does not reproduce the targeted synthetic facts verbatim and shows less structural paraphrasing in these examples, while preserving topic coherence and linguistic fluency. In retain scenarios, GU shows less over-forgetting and maintains semantic content, narrative structure, and reasoning ability, whereas baseline methods can distort or hallucinate non-target knowledge. These examples are illustrative rather than conclusive, and complement the aggregate quantitative metrics.