

# GEOMETRIC-DISENTANGLEMENT UNLEARNING

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Machine unlearning, the removal of a training subset’s influence from a deployed model, is critical for privacy preservation and model reliability, yet gradient ascent on forget samples often harms retained knowledge. Existing approaches face a persistent tradeoff between effective forgetting and preservation on the retain set. While previous methods provide useful heuristics, they often lack a formal analysis on how exactly forgetting updates harm retained knowledge, and whether the side effects can be removed with theoretical guarantees. To explore a theoretically sound and simple solution, we start from the first principle on *how performance on the retain set is actually affected*: a first-order analysis of the local change of the retain loss under small parameter updates during model training. We start from a crisp equivalence: the retain loss is unchanged to first order iff the update direction is orthogonal to the subspace spanned by retain gradients (“retain-invariant”). This identifies the entangled component as the tangential part of forget update within the retain-gradient subspace, and characterizes disentanglement as orthogonality. Guided by this, we propose the Geometric-disentanglement Unlearning (GU) that decomposes any candidate forget gradient update into tangential and normal components to retain space and executes only the normal component. Under a standard trust-region budget, the projected direction aligned with the raw forget gradient is optimal among all first-order retain-invariant moves, and we also derive the optimal projected direction for joint forget-retain updating objectives. Our method is plug-and-play and can be attached to existing gradient-based unlearning procedures to mitigate side effects. GU achieves consistent improvement on various methods across three benchmarks TOFU, MUSE, and WMDP. Plugging GU into SimNPO yields up to 62% lower Extraction Strength (ES), 32% higher retention ES, 8% higher utility, and 60% higher MIA-closeness on TOFU benchmark.

## 1 INTRODUCTION

Large language models learn broad knowledge from massive corpora (Touvron et al., 2023; Grattafiori et al., 2024; Wolf et al., 2020), but this strength also creates deployment risk: models can internalize private or harmful content that later must be removed (Carlini et al., 2021; Li et al., 2024a; Zhang et al., 2025). Machine unlearning aims to modify a trained model so that the influence of a forget set is erased while performance on the remaining retain data is preserved (Cao & Yang, 2015; Bourtole et al., 2021; Ginart et al., 2019; Graves et al., 2021). In practice, however, updates that improve forgetting often degrade behavior on retained content, revealing a persistent tradeoff between effective forgetting and retaining fidelity (Dorna et al., 2025; Maini et al., 2025; Chen & Yang, 2023; Yao et al., 2024). Existing approaches attempt to mitigate this tradeoff by incorporating empirical controls during fine-tuning (Dong et al., 2025; Yu et al., 2023; Ji et al., 2024) or by adjusting training preferences to balance the emphasis on forgetting and retaining data (Li et al., 2024b; Rafailov et al., 2023). While helpful in some cases, these strategies are often offline training-required (Bourtole et al., 2021; Ginart et al., 2019; Sendera et al., 2025), computationally heavy (Zhang et al., 2023; Bourtole & et al., 2021), or they rely on heuristic assumptions (Liu et al., 2024a) about why side effects arise, for example, attributing them to entanglement measured by embedding similarity (Long et al., 2024; Anonymous, 2025; Liu et al., 2024b; Xu et al., 2024b), without a formal, testable specification.

Hence, to derive a theoretically sound method with simple and accurate elimination of the tradeoff between forgetting and retaining, the central question is: exactly under what conditions does a

054 forgetting update cause side effects on retained knowledge, and can those effects be avoided with  
 055 theoretical guarantees? In fact, the “no side effect” condition yields a concrete standard for “retain-  
 056 invariant” updates that do not impact the performance of the retain set during training stage. It mo-  
 057 tivates us to explore the cause of side effects from a simple objective: **optimize** the forgetting loss  
 058 while leaving retained knowledge unchanged and enforcing this during training as a local “retain-  
 059 invariance” requirement. Rather than presupposing a representation of entanglement in embeddings  
 060 or parameters and mitigating it by heuristics, we first characterize which update directions leave  
 061 the retain loss locally unchanged. This analysis yields a concrete and testable account of the retain  
 062 forget interaction: the portion of an update that is responsible for first-order harm on retained data.  
 063 We prove a crisp equivalence: the retain loss is locally invariant if and only if the update direction is  
 064 orthogonal under the optimizer’s geometry to the subspace spanned by retain gradients. This charac-  
 065 terization identifies disentanglement with orthogonality to the retain gradient subspace. The tradeoff  
 066 arises from the tangential component of the forgetting update within this subspace, which perturbs  
 067 the retain loss, and a retain-invariant forgetting update should exclude this tangential component.

068 Motivated by this, we introduce Geometric-disentanglement Unlearning (GU). GU constructs the  
 069 orthogonal complement of the retain-gradient subspace and projects forgetting updates into that  
 070 complement before applying them, preserving only the normal component that leaves the retain loss  
 071 unchanged and removing the tangential interaction to reduce side effects. We show that, under a  
 072 standard trust-region budget, the projected direction most aligned with the raw forgetting gradient is  
 073 optimal and delivers the steepest descent progress while maintaining local invariance on the retain  
 074 knowledge. In addition, from an optimization perspective, we derive the optimal joint update direc-  
 075 tion of retain and forget gradients. Built on a simple and sound theoretical guarantee, GU integrates  
 076 easily into existing gradient-based unlearning pipelines: it only requires orthogonal projection from  
 forget to retain gradients, and does not alter core objectives or require additional regularizers.

077 Empirically, GU achieves stronger forgetting with smaller drift on the retain set, consistent with the  
 078 theoretical link between reduced entanglement and orthogonality-based retain-invariance. Specif-  
 079 ically, across three benchmarks using SimNPO (Fan et al., 2024), recognized as the SOTA  
 080 method (Dorna et al., 2025), adding our geometry-disentanglement projection yields up to 62%  
 081 lower forgetting Extraction Strength, 31% higher retention Extraction Strength, and 8% higher  
 082 model utility, and 60% higher MIA-closeness; on MUSE it cuts Extraction Strength (Carlini et al.,  
 083 2021) for Unlearning by 46%, boosts retained ROUGE by 17%, and reduces privacy-leak magnitude  
 084 by 14%; and on WMDP-cyber it lowers hazardous accuracy by 0.36% without harming MMLU.

085 Taken together, adopting orthogonality to the retain gradient subspace as an explicit design principle  
 086 provides a simple yet effective unified theoretical and practical framework for effective unlearning  
 087 with controlled side effects. Our contributions are threefold:

- 088 • We formalize and leverage a theoretically sound equivalence that local retain invariance matches  
 089 orthogonality to the retain-gradient subspace, thereby making side effects formally testable.
- 090 • We introduce Geometric-disentanglement Unlearning, a plug-and-play projection that provides a  
 091 simple and unified theoretical and practical framework for unlearning with controlled side effects.
- 092 • Across three well-known benchmarks of TOFU, MUSE, and WMDP, GU universally strengthens  
 093 forgetting while reducing collateral harm and preserving or improving downstream performance.

## 096 2 MACHINE UNLEARNING PRELIMINARIES

098 **Problem Definition.** Let  $\pi_\theta$  be the target model and  $\pi_{\text{ref}}$  be a reference model trained on a dataset  
 099  $D$ . Real-world data may contain private or harmful samples. Let  $D_f \subseteq D$  denote the forget subset  
 100 whose influence must be removed, and define the retain set  $D_r = D \setminus D_f$ . Starting from  $\pi_{\text{ref}}$ , we  
 101 continue training to obtain our model  $\pi_\theta$ . Our objective is for  $\pi_\theta$  to behave as if  $D_f$  had never been  
 102 used, which is to say, to match the behavior of a model trained from scratch on  $D_r$ . In principle, the  
 103 ideal approach is full retraining on  $D_r$ . However, in practice, this is often intractable due to heavy  
 104 costs. A common unlearning practice performs a bi-objective update at each step (Maini et al., 2024;  
 105 Dorna et al., 2025; Zhang et al., 2024). One samples a pair  $\{x_f, x_r\}$  with  $x_f \sim D_f$  and  $x_r \sim D_r$ .  
 106 The update applies **forget loss such as gradient ascent** on a forget objective evaluated at  $x_f$  and gra-  
 107 dient descent on a retain objective evaluated at  $x_r$ . The intent is to forget information associated with  
 $x_f$  while preventing unintended harm to  $x_r$ . We now make the two objectives explicit. Generally, for

108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161

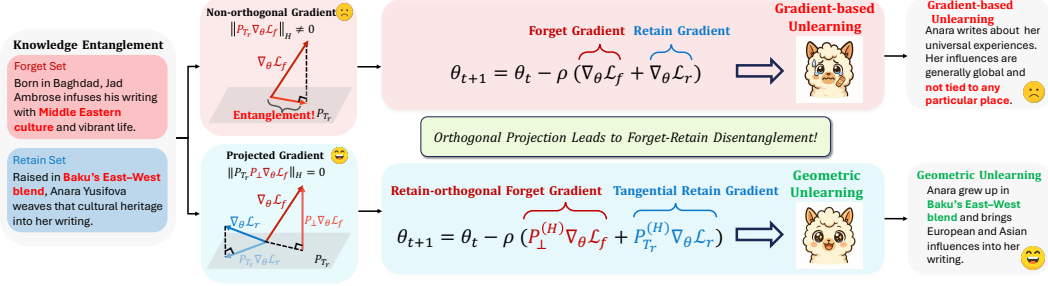


Figure 1: **Geometric Unlearning (bottom) vs. baseline (top)**.  $P_{\perp}$  is the  $H$ -orthogonal projector onto the complement of retain tangent subspace  $T_r$ ;  $P_{T_r}$  projects onto  $T_r$ . Without changing training objective or adding regularization, we *route* existing gradients through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

both the forget and retain training strategies, there are many viable choices. Taking forget loss as an example, we consider the following instantiations. *Token-level NLL*:  $\ell_f(x_f; \theta) \equiv$  sequence-averaged cross-entropy on  $x_f$  (with a sign conventionally chosen for ascent/descent as needed). *Preference ratios* (e.g., *SimNPO* (Fan et al., 2024)/*NPO* (Zhang et al., 2024)/*DPO* (Xu et al., 2024a)): use log-likelihood ratios against a frozen reference model  $\pi_{\text{ref}}$  to penalize the originally preferred response and/or promote an alternative. *Calibration-based variants* (e.g., *CEU* (Yang, 2025)/*UNDIAL* (Dong et al., 2025)/*WGA* (Wang et al., 2025)/*Sat-Imp* (Yang et al., 2025)): reshape logits or labels to discourage reproducing forget content. The loss can be instantiated in multiple ways, here we adopt the token-level NLL loss in the following practice for simplicity:

**Forget loss.** For a forget sample  $x_f \in D_f$ , let  $\ell_f(x_f; \theta)$  denote a *forget* objective that encourages the model to reject behaviors tied to  $D_f$ , optimized via *gradient ascent*:

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

**Retain loss.** Let  $\ell_r(x_r; \theta)$  denote a *retain* objective 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 will form training objectives that combine (i) a forget term aggregated over  $D_f$  and (ii) the retain-anchor  $L_r$ . A generic empirical objective takes the form  $\mathcal{L}_{\text{joint}}(\theta) = L_f(\theta) + \alpha L_r(\theta)$ , where  $\alpha \geq 0$  balances forgetting and retention.

### 3 METHODOLOGY

#### 3.1 GEOMETRIC-DISENTANGLEMENT UNLEARNING (GU)

**Featuring Side Effects on Retain Set.** Let  $\theta \in \mathbb{R}^p$  denote the parameters of the model  $\pi_{\theta}$ . Let  $D_f$  and  $D_r$  be the forget and retain sets, and let  $L_r : \mathbb{R}^p \rightarrow \mathbb{R}$  be the retain loss evaluated on  $D_r$ . 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” via heuristics without theoretically deriving a formal, testable specification of what is being disentangled (Liu et al., 2025a; Sendera et al., 2025). In contrast, to derive a theoretically rigorous forget-retain entanglement representation to mitigate the unlearning tradeoff accurately, we take a different route: starting from the desideratum *forget reduced, retain unchanged*, which manifests during training as a *local retain-invariance* requirement. Rather than presupposing a particular representation of entanglement (e.g., similarity in hidden states or in gradients), we first characterize the update directions that leave  $L_r$  locally invariant. This characterization, in turn, induces a theoretically grounded representation of entanglement, namely, the component of an update that is accountable for the harm on  $D_r$ . Concretely, during model training on a paired mini-step with forget and retain samples  $\{x_f, x_r\}$  at iteration  $t$ , a parameter update is written as  $\theta_{t+1} = \theta_t + \Delta\theta$ , where  $\Delta\theta \in \mathbb{R}^p$  is the step induced by the current optimization move. When  $\|\Delta\theta\|$  is small, retain loss  $L_r$ ’s local change at  $\theta_t$  along  $\Delta\theta$  admits the first order approximation:

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

where at training iteration  $t$ , we freeze an optimizer-induced symmetric positive definite (SPD) preconditioner  $H_t \succ 0$  and equip  $\mathbb{R}^p$  with the inner product  $\langle u, v \rangle_{H_t} := u^\top H_t v$  and norm  $\|v\|_{H_t} := \sqrt{\langle v, v \rangle_{H_t}}$ . Within the iteration, including any line-search or trust-region computation,  $H_t$  is treated as constant; it may be updated to  $H_{t+1}$  at the next step.<sup>1</sup> 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 may omit the  $t$  w.r.t  $H$ . i.e., under the metric  $H$ ,  $\nabla_\theta L_r(x_r, \theta_t)$  is the gradient of retain sample  $x_r$ . Our objective is to reduce this harm Eq. 3, ideally keeping  $L_r$  unchanged, which at the local scale means enforcing  $\Delta^{(1)} L_r = 0$ . To investigate when  $\Delta^{(1)} L_r = 0$ , we propose the following proposition:

**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  $\ell_r(x_r; \theta)$  be differentiable and define retain gradient  $g(x_r) := \nabla_\theta \ell_r(x_r; \theta) \in \mathbb{R}^p$ , 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  $L_r(\theta) := \sum_{i=1}^m \ell_r(x_{r_i}; \theta)$ ,  $\nabla_\theta L_r(\theta) = \sum_{i=1}^m g(x_{r_i}) \in T_r(\theta)$ . Then for any update direction  $\Delta\theta \in \mathbb{R}^p$ , the following are equivalent:

$$(i) \Delta\theta \in T_r(\theta)^\perp. \quad (ii) \Delta^{(1)} L_r(\theta; \Delta\theta) := \langle \nabla_\theta L_r(\theta), \Delta\theta \rangle_H = 0 \text{ for all } x_r \in D_r.$$

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. Proof see Appendix C.1.

**Geometric decomposition.** As established above, if an update direction is  $H$ -orthogonal to the retain gradient subspace, then the retain loss  $L_r$  is locally unchanged. This motivates a *geometric* view: as shown in Fig. 1, for a forget sample  $x_f$ , decompose its gradient

$$g_f(x_f) = P_{T_r}^{(H)} g_f(x_f) + P_\perp^{(H)} g_f(x_f), \quad (4)$$

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

$$P_{T_r}^{(H)} = U (U^\top H U)^{-1} U^\top H, \quad P_\perp^{(H)} = I - P_{T_r}^{(H)}, \quad (5)$$

where  $U = [u_1, \dots, u_k] \in \mathbb{R}^{p \times k}$  is the retain gradients from a small retain mini-batch  $B_r \subset D_r$  on selected tensors, spans the retain gradient subspace  $T_r = \text{range}(U)$  with  $U^\top H U = I$ .  $I$  is the  $k \times k$  identity matrix. The normal component  $P_\perp^{(H)}$  produces no first-order change on  $L_r$ , while the tangential component  $P_{T_r}^{(H)}$  captures the interaction with retain updates. We therefore define retain-forget gradient update entanglement by the magnitude of the tangential component:

$$\text{ent}_H(g_f(x_f)) := \|P_{T_r}^{(H)} g_f(x_f)\|_H, \quad (6)$$

which vanishes if and only if  $g_f(x_f) \in T_r^\perp$ . In the disentangled case,  $P_\perp^{(H)} g_f(x_f)$  yields a direction that is first-order safe for  $L_r$ . However, the retain-invariance subspace  $T_r^\perp$  contains infinite directions. Which retain-invariance direction should we take under a fixed step budget for local optimal forgetting? A first-order selection principle can answer this. Fix the metric  $H \succ 0$ , first-order linearizing the joint objective  $\mathcal{L}_{\text{joint}}(\theta) := L_f(\theta) + \alpha L_r(\theta)$  at  $\theta_t$  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*, the steepest first-order change of  $\mathcal{L}_{\text{joint}}$  coincides with that of  $L_f$ , and depends only on  $g_f := \nabla^H L_f(\theta_t)$  projected onto  $T_r^\perp$ . This leads to the following lemma:

<sup>1</sup>This 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 (2014); natural-gradient/K-FAC metrics (Amari et al., 2019; Martens & Grosse, 2015)).

**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*

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

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

$$\Delta\theta_f^* = \arg \min_{\Delta\theta \in \mathcal{C}} \langle g_f, \Delta\theta \rangle_H = - \frac{P_\perp^{(H)} g_f}{\|P_\perp^{(H)} g_f\|_H}, \quad \text{unique if } P_\perp^{(H)} g_f \neq 0.$$

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  $\mathcal{C}$ :

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

Proof see Appendix C.2 for details. This provides the optimal update step for the total loss  $\mathcal{L}_{\text{joint}}$ :

$$\theta_{t+1} = \theta_t - \rho \left( \underbrace{P_\perp^{(H)} \nabla^H L_f(\theta_t)}_{\text{retain-orthogonal}} + \underbrace{P_{T_r}^{(H)} \nabla^H L_r(\theta_t)}_{\text{retain-tangent}} \right) \quad (7)$$

is first-order optimal for the joint objective under the retain-safety constraint. Under standard  $H$ -smoothness, the step size  $\rho$  can be selected by a trust-region or line-search rule (see § 3.2, Proposition 3.3, Corollary 3.4). Noted that our geometric-disentanglement update is a plug-and-play method, and projection touches only selected trainable tensors, making GU architecture-agnostic. We present algorithm details in Appendix B for the basis calculation and the optimizer update step.

### 3.2 THEORETICAL GUARANTEES

We provide the theoretical guarantees for our method, GU. We have proven that  $P_\perp^{(H)} g_f$  is the *steepest safe* direction for  $\mathcal{L}_{\text{joint}}$  in Lemma 3.2, furthermore, we will show  $L_r$  is *first-order nonincreasing* (strictly decreasing when  $\beta > 0$ ), with second-order drift bounded by smoothness in Prop. 3.3 and its corollary Cor. 3.4). Then, we show that the composite objective enjoys a *nonpositive* first-order change with an explicit negative lower bound in Prop. 3.5. Collectively, these results justify GU as a principled *first-order safe* and *steepest-feasible* unlearning procedure in the optimizer geometry, with explicit stability and robustness margins.

**First-Order Safety and Retain Monotonicity** The next proposition quantifies, at first order, how this step impacts the retain loss  $L_r$ : the normal forget component is first-order neutral to  $L_r$ , whereas the tangential repair strictly decreases  $L_r$  whenever  $g_r \neq 0$ .

**Proposition 3.3** (First-order safety and retain monotonicity). *Let  $H \succ 0$  be SPD and let  $T_r \subset \mathbb{R}^p$  denote the retain-gradient subspace w.r.t. the  $H$ -inner product. Let  $g_r := \nabla_\theta^H L_r(\theta_t) \in T_r$  and  $g_f := \nabla_\theta^H L_f(\theta_t)$ . WLOG, introduce  $\beta \geq 0$ . Consider one split step*

$$\Delta\theta = -\rho \left( P_\perp^{(H)} g_f + \beta P_{T_r}^{(H)} g_r \right) \quad \text{with } \rho > 0, \beta \geq 0. \quad (8)$$

Then the first-order change of  $L_r$  satisfies

$$\Delta^{(1)} L_r = \langle g_r, \Delta\theta \rangle_H = -\rho \beta \|g_r\|_H^2 \leq 0. \quad (9)$$

If  $\beta = 0$  the step is first-order neutral to  $L_r$ , and if  $\beta > 0$ ,  $g_r \neq 0$  it is first-order strictly decreasing.

Proposition 3.3 establishes the *first-order* effect of one split step on the retain loss:  $\Delta^{(1)} L_r = \langle g_r, \Delta\theta \rangle_H = -\rho \beta \|g_r\|_H^2 \leq 0$ . To convert this into an *actual* decrease of  $L_r(\theta)$ , we invoke the  $H$ -geometry version of the descent lemma under Lipschitz  $H$ -gradient, and combine it with the  $H$ -orthogonal decomposition of the step:

**Corollary 3.4** (Descent guarantee for  $L_r$  under  $H$ -smoothness). *Assume the  $H$ -gradient  $\nabla_\theta^H L_r$  is  $L_r^{(H)}$ -Lipschitz under  $\|\cdot\|_H$ , i.e.,  $\|\nabla_\theta^H L_r(\theta + \Delta) - \nabla_\theta^H L_r(\theta)\|_H \leq L_r^{(H)} \|\Delta\|_H$ . Let the split step be  $\Delta\theta = -\rho(P_\perp^{(H)} g_f + \beta P_{T_r}^{(H)} g_r)$  with  $\rho > 0$  and  $\beta \geq 0$ , where  $g_r := \nabla_\theta^H L_r(\theta)$  and  $g_f := \nabla_\theta^H L_f(\theta)$ . Then*

$$L_r(\theta + \Delta\theta) \leq L_r(\theta) - \rho \beta \|g_r\|_H^2 + \frac{L_r^{(H)}}{2} \rho^2 \left( \|P_\perp^{(H)} g_f\|_H^2 + \beta^2 \|g_r\|_H^2 \right). \quad (10)$$

In particular, if  $0 < \rho < \frac{2\beta\|g_r\|_H^2}{L_r^{(H)}(\|P_\perp^{(H)}g_f\|_H^2 + \beta^2\|g_r\|_H^2)}$ , then  $L_r(\theta + \Delta\theta) < L_r(\theta)$  (strict descent whenever  $\beta > 0$  and  $g_r \neq 0$ ). For  $\beta = 0$ ,

$$L_r(\theta + \Delta\theta) \leq L_r(\theta) + \frac{L_r^{(H)}}{2}\rho^2\|P_\perp^{(H)}g_f\|_H^2 = L_r(\theta) + O(\rho^2), \quad (11)$$

recovering the neutral first-order case with only second-order drift.

Proof details of Proposition 3.3 and Corollary 3.4 in Appendix C.3.

**One-step behavior of the joint objective.** Having established first-order monotonicity and actual descent for  $L_r$ , we now analyze the one-step first-order change of the joint objective  $\mathcal{L}_{\text{joint}} := L_f + \alpha L_r$  under the same split step.

**Proposition 3.5** (Exact first-order change of  $\mathcal{L}_{\text{joint}}$ ). *Let  $H \succ 0$ ,  $g_f := \nabla_\theta^H L_f(\theta)$ ,  $g_r := \nabla_\theta^H L_r(\theta) \in T_r$ , and  $\Delta\theta = -\rho(P_\perp^{(H)}g_f + \beta P_{T_r}^{(H)}g_r)$  with  $\rho > 0$ ,  $\beta \geq 0$ . Then the first-order change of the joint objective equals*

$$\Delta^{(1)}\mathcal{L}_{\text{joint}} := \langle g_f + \alpha g_r, \Delta\theta \rangle_H = -\rho\left(\|P_\perp^{(H)}g_f\|_H^2 + \alpha\beta\|g_r\|_H^2 + \beta\langle P_{T_r}^{(H)}g_f, g_r \rangle_H\right). \quad (12)$$

Proof details see Appendix C.4. In the optimizer-induced metric  $H$ , we prove that GU performs first-order-safe, steepest-feasible forgetting by projecting onto the retain-orthogonal subspace, guarantees monotone decrease of the retain loss via an explicit stepsize condition, provides an exact one-step decomposition for the joint objective with verifiable nonpositivity conditions, and quantifies retain-forget entanglement by the norm of the tangential component  $\|P_{T_r}^{(H)}g_f\|_H$ .

## 4 EXPERIMENTS

### 4.1 SETTINGS

**Datasets** We evaluate our method on the **OpenUnlearning** benchmark suite, focusing primarily on **TOFU** (Dorna et al., 2025), a fine-grained benchmark with 200 fictitious author profiles, each containing 20 QA pairs. For fair comparison, we adopt the Llama-3 backbones (1B, 3B, 8B) (Dubey et al., 2024) provided by the suite and follow the official *scaling splits*, varying the forget set size (`forget01`, `forget05`, `forget10`) to examine scalability. In addition, we report results on **MUSE** (Shi et al., 2025), which evaluates memorization and unlearning of books and news articles through verbatim reproduction, question answering, and membership inference, and on **WMDP** (Liu et al., 2024c), an alignment-oriented benchmark of 3,668 multiple-choice questions across hazardous domains (biosecurity, cybersecurity, chemical security) assessing whether models can forget dangerous capabilities while retaining general performance. For MUSE and WMDP, we report results on Llama-2-7B (Touvron et al., 2023) and zephyr-7b (Tunstall et al., 2024) to provide a more comprehensive evaluation<sup>2</sup>.

**Evaluation Metrics** Following Dorna et al. (2025); Yang et al. (2025), we evaluate unlearning performance along four axes. **Forgetting** is measured by *Extraction Strength on the forget set* (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 *Extraction Strength (Carlini et al., 2021) on the retain set* (ES,Re.; ↑), monitoring collateral damage to preserved knowledge; for MUSE and WMDP, this is complemented with *ROUGE RE*, which measures generation quality on retained knowledge-based QA pairs. **Privacy** is captured by resistance to membership inference: on TOFU we report *MIA closeness* (↑), which evaluates the similarity between unlearned and retain-only models across multiple MIA variants, while on MUSE and WMDP we use *Privacy Leakage (Priv. Leak.; ↓)*, which directly tests whether membership information from the forget set can still be inferred. Finally, **Utility** (↑) captures post-unlearning usefulness: TOFU reports a composite model-utility score combining probability, ROUGE, and Truth Ratio across retain and factual knowledge sets, while MUSE

<sup>2</sup>These models are chosen to ensure they have learned the target knowledge, enabling fair comparison of unlearning effectiveness, as provided and recommended in the OpenUnlearning suite.

Table 1: TOFU results comparing unlearning objectives with a retain-oriented geometric regularizer (GU) that stabilizes updates via retain-null projection. Arrows  $\uparrow$  /  $\downarrow$  denote that higher/lower is better. Within each block (model scale and deletion rate), the top two entries are shaded: **blue** for higher-is-better metrics and **red** for lower-is-better metrics (no boldface is used). Abbreviations: ES Re. = Extraction Strength on the retain split; ES Un. = Extraction Strength on the forget split; Priv. = MIA closeness; MU = composite model utility. “Forget-1%, 5%, 10%” indicate the fraction of TOFU authors deleted. “Vanilla” is the pretrained backbone without TOFU fine-tuning; “fully-finetuned” is trained on the full TOFU corpus. “w. GU” denotes the corresponding objective augmented with our geometry module.

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	0.0649	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

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). A detailed introduction of the metrics refer to open-unlearning (Dorna et al., 2025).

#### 4.2 GEOMETRY-GUIDED UNLEARNING DELIVERS PARETO IMPROVEMENTS ON TOFU

Table 1 shows the result on TOFU benchmark. We use two informative references. *Vanilla* is the pretrained backbone without any TOFU fine-tuning; it neither learns nor regurgitates TOFU facts

Table 2: MUSE benchmark results for Llama-2-7b-hf on Books and News splits and WMDP benchmark results for zephyr-7b-beta on cyber split.  $\downarrow$  indicates smaller values are better, while  $\uparrow$  indicates larger values are better.

Method	MUSE Books			MUSE News			WMDP cyber	
	ES Un. $\downarrow$	Priv. Leak. $\rightarrow 0$	ROUGE Re. $\uparrow$	ES Un. $\downarrow$	Priv. Leak. $\rightarrow 0$	ROUGE Re. $\uparrow$	Un. acc. $\downarrow$	mmlu acc. $\uparrow$
	<b>Llama-2-7b-hf</b>						<b>zephyr-7b-beta</b>	
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

(low ES on both splits), enjoys perfect MIA-closeness ( $\text{Priv} = 1$ ), and yields moderate utility. *Fully-finetuned* is trained on the entire TOFU corpus; it memorizes broadly (high ES on both splits), collapses privacy ( $\text{Priv} = 0$ ), and reaches a utility ceiling. The practical goal is to move unlearning methods off these single-orbit 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.

**Observed Pareto shifts at fixed or lower ES Un.** In Figure 2, the Base  $\rightarrow$  GU shifts consistently follow a Pareto-improving direction across ES all metrics. Specifically, (i) ES Un decreases further, indicating that GU not only preserves but even slightly improves forgetting effectiveness; (ii) ES Re and Priv increase, demonstrating that GU substantially mitigates the trade-off typically observed in existing unlearning methods, forgetting the target knowledge no longer harms retained knowledge or privacy; and (iii) MU remains stable or improves, showing that GU enhances unlearning without compromising overall model utility. Taken together, these trends highlight that GU enables precise and low-side-effect unlearning, transforming the unlearning process from a severe trade-off challenge into a near-Pareto-optimal operation. Specifically, across Llama-3.2 at 1B/3B, Llama-3.1 8B and unlearning rates `forget01/05/10`, adding geometry-disentanglement projection (“w.GU”) to diverse objectives reliably raises ES on the retain split and improves utility without worsening ES on the forget split. Three representative cases illustrate the pattern. (i) CEU at 1B and `forget01`: ES,Re increases (0.0875 $\rightarrow$ 0.2236) and MU rises (0.3666 $\rightarrow$ 0.5134) while ES,Un stays near the floor (0.0316 $\rightarrow$ 0.0328). (ii) UNDIAL at 3B and `forget10`: ES,Re increases (0.3538 $\rightarrow$ 0.7869) and MU improves (0.6550 $\rightarrow$ 0.6992) with a slight decrease in ES,Un (0.0416 $\rightarrow$ 0.0396). (iii) SimNPO at 8B and `forget01`: ES,Un drops sharply (0.3101 $\rightarrow$ 0.1177) while ES,Re nudges upward (0.8256 $\rightarrow$ 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 consistently low (0.0658 $\rightarrow$ 0.0619 and 0.0416 $\rightarrow$ 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

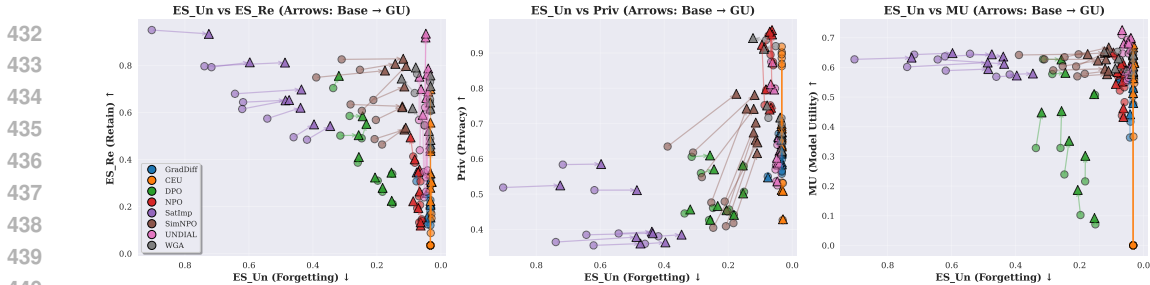


Figure 2: We visualize forgetting quality (ES Un: lower for better) against retained knowledge (ES Re), privacy (Priv), and model utility (MU) for eight unlearning baselines on TOFU. ES Re, Priv, and MU are metrics of higher for better. Circles denote baseline outputs, triangles denote results of GU, and arrows indicate the shift from Base  $\rightarrow$  GU. Across all three panels, GU pushes methods toward the Pareto-optimal corner (upper-right), reducing the trade-off between forgetting and retaining.

to 0.9526. For WGA, UNDIAL, and SimNPO, privacy is typically preserved or slightly improved while retention and utility rise, consistent with 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 whitened 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.

### 4.3 MUSE & WMDP: CONSISTENCY OF GEOMETRY-GUIDED IMPROVEMENTS.

We evaluate on two complementary settings: (i) *MUSE* (Llama-2-7B) probes verbatim reproduction and QA over Books/News, where forgetting should *reduce* ES on the forget split while *preserving* ROUGE on retained QA and *reducing* privacy leakage toward zero; (ii) *WMDP-cyber* (zephyr-7b-beta) probes capability removal, where lower unlearning accuracy (Un. acc.  $\downarrow$ ) signals safer behavior while general ability (MMLU  $\uparrow$ ) should not degrade. Table 2 shows the results on these two benchmarks. The Pareto improvement and detailed analysis is in Appendix D.3.

## 5 RELATED WORK

**LLM Unlearning and Orthogonal Decomposition.** Unlearning in LLMs seeks to remove specific data influence while preserving general performance. Approaches include *gradient-based unlearning*, which maximizes loss on target samples but risks catastrophic forgetting (Thudi et al., 2022; Izzo et al., 2021), and *preference-based optimization* like NPO, offering more stable updates via constrained objectives (Li et al., 2024b). *Parameter-efficient methods* modify adapters or LoRA layers to balance forgetting and retention (Yu et al., 2023; Kurmanji et al., 2023). *Representation-based techniques* remove knowledge via hidden state manipulation or teacher distillation (Cao & Yang, 2015; Ginart et al., 2019). *Inference-time methods* use prompts or embedding corruption for fast but superficial unlearning (Maini et al., 2025). However, traces of forgotten content often persist, e.g., adversarial prompts can recover them (Jagielski et al., 2023; Liu et al., 2024c), and outputs remain distinguishable (Chen et al., 2025). This highlights the gap between true and fake erasure, motivating our method for principled trade-offs between forgetting and retention without degrading real-world performance. *Overcoming catastrophic forgetting by gradient projection* (Kirkpatrick et al., 2017). *Gradient surgery for multi-task learning* (Yu et al., 2020). *Orthogonal gradient descent for continual learning* (Farajtabar et al., 2020). Recent work brings similar geometric control to machine unlearning: PGU (Hoang et al., 2024), UNSC (Chen et al., 2024), and SEMU (Sendera et al., 2025) construct projection or null-space updates from activations or SVDs, while NegMerge (Kim et al., 2024) and NatMU (He et al., 2025) design weight- and label-space edits, and Deep Unlearn (Cadet et al., 2024) benchmarks these heuristics. However, these methods target small supervised models or do not provide an optimizer-aware gradient orthogonalization that scales to LLMs. More detailed comparison and discussion refer to the Appendix A.4.

**Knowledge Conflict and Entanglement in LLMs.** LLMs face conflicts between internal memory and external evidence, categorized as *context-memory*, *inter-context*, and *intra-memory* conflicts (Xu et al., 2024b; Li et al., 2025). For context-memory conflicts, confidence metrics guide reliance on parametric vs. retrieved knowledge (Pang et al., 2024). *Knowledge editing* methods (e.g., ROME, MEMIT, FT-Edit) overwrite facts but risk interference. Recent methods like *AlphaEdit* and *GeoEdit* constrain updates via subspace projection to minimize side effects (Fang et al., 2025; Feng et al., 2025). Additional approaches include *disentanglement* (Long et al., 2024) and *consistency tuning* (Wang et al., 2024). Unresolved conflicts often cause models to default to internal memory, ignoring external input (Xu et al., 2024b). While sharing goals with editing-based methods, our approach offers principled control over forgetting-retention trade-offs with theoretical guarantees. Knowledge entanglement refers to interdependent representations where removing one piece disrupts others. TOFU shows widespread collateral forgetting (Maini et al., 2024); MicroEdit links interference to polysemantic neurons and proposes neuron-level edits (Anonymous, 2025); other work finds biased or memorized knowledge entangled with core reasoning (Liu et al., 2025b; Ghosal et al., 2025). These studies demonstrate that LLM knowledge is stored in distributed, overlapping forms, complicating precise unlearning. In the absence of provable disentanglement, current methods mainly rely on heuristics. *SEMU* and *Deep Unlearning* use SVD-based subspace isolation to confine forgetting (Sendera et al., 2025; Kodge et al., 2024). *UNSC* adjusts updates in the null space of retained knowledge (Chen et al., 2024); *ECO* avoids weight edits by corrupting prompts at inference (Liu et al., 2024b); *MicroEdit* sparsifies edits to monosemantic neurons (Anonymous, 2025). These strategies, via low-rank approximations, null-space constraints, sparse edits, or prompt manipulations, approximate disentanglement, though full theoretical guarantees remain open.

## 6 CONCLUSION

To seek a theoretically sound and simple solution that precisely reduces forget-retain tradeoff, we studied when forgetting updates leave retained knowledge unchanged and showed that a local retain-invariance requirement aligns with orthogonality to the retain-gradient subspace. Building on this equivalence, we introduced Geometric-disentanglement Unlearning, which projects updates onto the retain-orthogonal complement, reducing side effects on the retain set. GU is plug-and-play, optimizer-compatible, and architecture-agnostic, and it attaches seamlessly to existing gradient-based unlearning pipelines to mitigate collateral harm. Across TOFU, MUSE, and WMDP, GU delivers consistent improvements in forgetting while preserving or enhancing retained performance.

**Ethics Statement** Our work focuses on the ethical need to remove private, harmful, or unauthorized content from large language models while preserving legitimate capabilities. Unlearning is inherently privacy- and safety-focused because it touches how models retain or discard information about individuals and sensitive domains. Our method GU is designed to reduce unintended degradation on non-target content, which aligns with the goals of privacy protection, user trust, and reliable deployment. For data governance and human subjects, we use public benchmarks curated for research and do not introduce new personally identifiable information. Any future deployment should follow data minimization, consent, and legal compliance.

**Reproducibility Statement.** We emphasize reproducibility throughout the paper. § 3 presents the core algorithm and training workflow, and the 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, basis rank  $k$ , refresh period, residual threshold, projected layer range  $K$ , mixing weights  $(\gamma, \alpha)$ , trust-region parameters  $(\kappa, \tau)$ , optimizer choices, and learning-rate schedules, so readers can reproduce results without additional assumptions and readily extend our experiments. Our experiments run in a server with Intel Gold CPU with 1024 Gb Memory and 2 H100 GPU.

## REFERENCES

- 540  
541  
542 Shun-ichi Amari, Ryo Karakida, and Masafumi Oizumi. Fisher information and natural gradient  
543 learning in random deep networks. In *The 22nd International Conference on Artificial Intelligence*  
544 *and Statistics*, pp. 694–702. PMLR, 2019.
- 545  
546 Anonymous. Microedit: Neuron-level knowledge disentanglement and localization in lifelong  
547 model editing. In *OpenReview preprint*, 2025. URL [https://openreview.net/forum?](https://openreview.net/forum?id=FDZurhzV3y)  
548 [id=FDZurhzV3y](https://openreview.net/forum?id=FDZurhzV3y). Under review.
- 549  
550 Lucas Bourtole, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin  
551 Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE*  
552 *symposium on security and privacy (SP)*, pp. 141–159. IEEE, 2021.
- 553  
554 Ludovic Bourtole and et al. Machine unlearning. In *IEEE Symposium on Security and Privacy*,  
555 2021.
- 556  
557 Xavier F Cadet, Anastasia Borovykh, Mohammad Malekzadeh, Sara Ahmadi-Abhari, and Hamed  
558 Haddadi. Deep unlearn: Benchmarking machine unlearning. *arXiv preprint arXiv:2410.01276*,  
559 2024.
- 560  
561 Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE*  
562 *Symposium on Security and Privacy*, 2015.
- 563  
564 Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine  
565 Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data  
566 from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp.  
567 2633–2650, 2021.
- 568  
569 Huiqiang Chen, Tianqing Zhu, Xin Yu, and Wanlei Zhou. Machine unlearning via null space calibra-  
570 tion. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI*  
571 *2024)*, 2024. URL <https://www.ijcai.org/proceedings/2024/0040.pdf>.
- 572  
573 Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. In  
574 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*  
575 *(EMNLP)*, pp. 12041–12052, 2023.
- 576  
577 Zehong Chen, Yuxin Wu, Yuchen Ma, et al. Footprints of unlearning in large language models.  
578 *arXiv preprint arXiv:2503.04521*, 2025.
- 579  
580 Yijiang River Dong, Hongzhou Lin, Mikhail Belkin, Ramon Huerta, and Ivan Vulić. UNdIAL: Self-  
581 distillation with adjusted logits for robust unlearning in large language models. In *Proceedings*  
582 *of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computa-*  
583 *tional Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8827–8840,  
584 Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-  
585 8-89176-189-6. URL <https://aclanthology.org/2025.naacl-long.444/>.
- 586  
587 Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zachary C Lipton, J Zico  
588 Kolter, and Pratyush Maini. Openunlearning: Accelerating llm unlearning via unified bench-  
589 marking of methods and metrics. *arXiv preprint arXiv:2506.12618*, 2025.
- 590  
591 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha  
592 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models.  
593 *arXiv e-prints*, pp. arXiv–2407, 2024.
- 594  
595 John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and  
596 stochastic optimization. *Journal of machine learning research*, 12(7), 2011.
- 597  
598 Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Sim-  
599 plicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Neurips*  
600 *Safe Generative AI Workshop 2024*, 2024. URL [https://openreview.net/forum?id=](https://openreview.net/forum?id=pVACX02m0p)  
601 [pVACX02m0p](https://openreview.net/forum?id=pVACX02m0p).

- 594 Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Shi Jie, Xiang Wang, Xiangnan He, and  
595 Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. In  
596 *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. doi: 10.  
597 48550/arXiv.2410.02355. URL <https://arxiv.org/abs/2410.02355>. Oral.
- 598 Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for contin-  
599 ual learning. In *International conference on artificial intelligence and statistics*, pp. 3762–3773.  
600 PMLR, 2020.
- 601 Yujie Feng, Liming Zhan, Zexin Lu, Yongxin Xu, Xu Chu, Yasha Wang, Jiannong Cao, Philip S.  
602 Yu, and Xiao-Ming Wu. Geoedit: Geometric knowledge editing for large language models. *arXiv*  
603 *preprint arXiv:2502.19953*, 2025. doi: 10.48550/arXiv.2502.19953. URL <https://arxiv.org/abs/2502.19953>.
- 604 Gaurav R. Ghosal, Pratyush Maini, and Aditi Raghunathan. Memorization sinks: Isolating mem-  
605 orization during llm training. In *Proceedings of the 42nd International Conference on Machine*  
606 *Learning (ICML)*, 2025. URL <https://arxiv.org/abs/2507.09937>. Also available as  
607 arXiv:2507.09937.
- 608 Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data  
609 deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.
- 610 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad  
611 Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of  
612 models. *arXiv preprint arXiv:2407.21783*, 2024. URL <https://arxiv.org/abs/2407.21783>.
- 613 Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of*  
614 *the AAAI Conference on Artificial Intelligence*, number 13, pp. 11516–11524, 2021.
- 615 Zhengbao He, Tao Li, Xinwen Cheng, Zhehao Huang, and Xiaolin Huang. Towards Natural Ma-  
616 chine Unlearning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 47(12):  
617 11548–11560, December 2025. ISSN 1939-3539. doi: 10.1109/TPAMI.2025.3597350. URL  
618 <https://doi.ieeecomputersociety.org/10.1109/TPAMI.2025.3597350>.
- 619 Tuan Hoang, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Learn to unlearn for deep neural  
620 networks: Minimizing unlearning interference with gradient projection. In *Proceedings of the*  
621 *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4819–4828, 2024.
- 622 Zachary Izzo, Aaron Smart, James Zou, et al. Approximate data deletion from machine learning  
623 models. In *ICML*, 2021.
- 624 Matthew Jagielski, Nicholas Carlini, Milad Nasr, et al. Tofu: Benchmarking machine unlearning for  
625 large language models. *arXiv preprint arXiv:2310.10683*, 2023.
- 626 Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana Kompella, Sijia Liu, and Shiyu Chang. Re-  
627 versing the forget-retain objectives: An efficient llm unlearning framework from logit difference.  
628 *Advances in Neural Information Processing Systems*, 37:12581–12611, 2024.
- 629 Hyoseo Kim, Dongyoon Han, and Junsuk Choe. Negmerge: Consensual weight negation for strong  
630 machine unlearning. *arXiv preprint arXiv:2410.05583*, 2024.
- 631 Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*,  
632 2014.
- 633 James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A  
634 Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcom-  
635 ing catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*,  
636 114(13):3521–3526, 2017.
- 637 Sangamesh Kodge, Gobinda Saha, and Kaushik Roy. Deep unlearning: Fast and efficient gradient-  
638 free class forgetting. *Transactions on Machine Learning Research*, 2024. URL <https://arxiv.org/abs/2312.00761>. TMLR (July 2024).

- 648 Mohammad Kurmanji, Eleni Triantafillou, Shuangchi Wang, et al. Towards machine unlearning in  
649 large neural networks. *arXiv preprint arXiv:2303.07258*, 2023.
- 650
- 651 Gaotang Li, Yuzhong Chen, and Hanghang Tong. Taming knowledge conflicts in language models.  
652 *arXiv preprint arXiv:2503.10996*, 2025.
- 653 Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D.  
654 Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin  
655 Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xi-  
656 aoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B Breuer, Andy  
657 Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam Alfred Hunt, Justin  
658 Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad  
659 Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar,  
660 Uday Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr  
661 Wang, and Dan Hendrycks. The wmdp benchmark: Measuring and reducing malicious use with  
662 unlearning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*,  
663 volume 235 of *Proceedings of Machine Learning Research*, pp. 28525–28550. PMLR, 2024a.  
664 URL <https://proceedings.mlr.press/v235/li24bc.html>.
- 665 Yifan Li, Yilun Chen, Kai Zhang, et al. Negative preference optimization for unlearning in large  
666 language models. *arXiv preprint arXiv:2406.09228*, 2024b.
- 667
- 668 Chris Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. Large language model unlearning via  
669 embedding-corrupted prompts. *Advances in Neural Information Processing Systems*, 37:118198–  
670 118266, 2024a.
- 671 Chris Yuhao Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. Large language model unlearning  
672 via embedding-corrupted prompts. *arXiv preprint arXiv:2406.07933*, 2024b. URL <https://arxiv.org/abs/2406.07933>.
- 673
- 674 Zheyuan Liu, Suraj Maharjan, Fanyou Wu, Rahil Parikh, Belhassen Bayar, Srinivasan H. Sen-  
675 gamedu, and Meng Jiang. Disentangling biased knowledge from reasoning in large language  
676 models via machine unlearning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and  
677 Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association  
678 for Computational Linguistics (Volume 1: Long Papers)*, pp. 6105–6123, Vienna, Austria, July  
679 2025a. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/  
680 2025.acl-long.305. URL <https://aclanthology.org/2025.acl-long.305/>.
- 681 Zheyuan Liu, Suraj Maharjan, Fanyou Wu, Rahil Parikh, Belhassen Bayar, Srinivasan H. Sen-  
682 gamedu, and Meng Jiang. Disentangling biased knowledge from reasoning in large language  
683 models via machine unlearning. In *Proceedings of the 63rd Annual Meeting of the Associa-  
684 tion for Computational Linguistics (Volume 1: Long Papers)*, pp. 6105–6123, Vienna, Austria,  
685 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.305. URL  
686 <https://aclanthology.org/2025.acl-long.305/>.
- 687
- 688 Zijie Liu, Yihan Wang, Shuo Zhang, et al. Wmdp: A benchmark for evaluating machine unlearning  
689 in language models. *arXiv preprint arXiv:2402.08082*, 2024c.
- 690
- 691 Chen Long, Shuo Wang, and Yifan Liu. Disentangling conflicting knowledge in large language  
692 models. *arXiv preprint arXiv:2405.12345*, 2024.
- 693
- 694 Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. TOFU: A  
695 task of fictitious unlearning for LLMs. *First Conference On Language Modeling*, 2024. URL  
696 <https://openreview.net/pdf?id=B4lhNB0WLo>.
- 697
- 698 Pratyush Maini, Matthew Jagielski, Florian Tramèr, et al. Unlearning in large language models: A  
699 survey. *arXiv preprint arXiv:2503.01854*, 2025.
- 700
- 701 James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate  
curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.
- Xinyu Pang, Ziyu Zhang, Tian Wang, et al. Detecting and resolving context-memory conflicts in  
large language models. *arXiv preprint arXiv:2402.04562*, 2024.

- 702 Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea  
703 Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances*  
704 *in Neural Information Processing Systems*, 36:53728–53741, 2023.
- 705  
706 Marcin Sendera, Lukasz Struski, Kamil Ksikazek, Kryspin Musiol, Jacek Tabor, and Dawid Ry-  
707 marczyk. Semu: Singular value decomposition for efficient machine unlearning. *arXiv preprint*  
708 *arXiv:2502.07587*, 2025. URL <https://arxiv.org/abs/2502.07587>.
- 709 Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao  
710 Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. MUSE: Machine unlearning six-  
711 way evaluation for language models. In *The Thirteenth International Conference on Learning*  
712 *Representations*, 2025. URL <https://openreview.net/forum?id=TArmA033BU>.
- 713  
714 Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Un-  
715 derstanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on*  
716 *Security and Privacy (EuroS&P)*, pp. 303–319. IEEE, 2022.
- 717 Hugo Touvron, Marco Minervini, Emma Chi, Arturo Figueroa, et al. Llama: Open and efficient  
718 foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- 719  
720 Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul,  
721 Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourier, Nathan Habib,  
722 Nathan Sarrazin, Omar Sanseviero, Alexander M Rush, and Thomas Wolf. Zephyr: Di-  
723 rect distillation of LM alignment. In *First Conference on Language Modeling*, 2024. URL  
724 <https://openreview.net/forum?id=aKkAwZB6JV>.
- 725  
726 Jialin Wang, Hao Zhou, Jian He, et al. Consistency fine-tuning for mitigating knowledge conflicts  
727 in llms. *arXiv preprint arXiv:2407.06789*, 2024.
- 728  
729 Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q Weinberger.  
730 Rethinking llm unlearning objectives: A gradient perspective and go beyond. *arXiv preprint*  
731 *arXiv:2502.19301*, 2025.
- 732  
733 Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi,  
734 Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick  
735 von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gug-  
736 ger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art  
737 natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in*  
738 *Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. As-  
739 sociation for Computational Linguistics. URL [https://www.aclweb.org/anthology/](https://www.aclweb.org/anthology/2020.emnlp-demos.6)  
740 [2020.emnlp-demos.6](https://www.aclweb.org/anthology/2020.emnlp-demos.6).
- 741  
742 Shusheng Xu, Wei Fu, Jiakuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu,  
743 and Yi Wu. Is DPO superior to PPO for LLM alignment? a comprehensive study. *arXiv preprint*  
744 *arXiv:2404.10719*, 2024a.
- 745  
746 Zhiqiang Xu, Rui Zhang, Xiaoyu Chen, et al. Survey of knowledge conflicts in large language  
747 models. *arXiv preprint arXiv:2401.12129*, 2024b.
- 748  
749 Bo Yang. CE-U: Cross Entropy unlearning. *arXiv preprint arXiv:2503.01224*, 2025.
- 750  
751 Puning Yang, Qizhou Wang, Zhuo Huang, Tongliang Liu, Chengqi Zhang, and Bo Han. Exploring  
752 criteria of loss reweighting to enhance llm unlearning. *arXiv preprint arXiv:2505.11953*, 2025.
- 753  
754 Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, and Tianhao Wang. Machine unlearning of pre-trained  
755 large language models. *arXiv preprint arXiv:2402.15159*, 2024.
- 756  
757 Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and  
758 Chelsea Finn. Gradient surgery for multi-task learning. In H. Larochelle, M. Ran-  
759 zato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information*  
760 *Processing Systems*, volume 33, pp. 5824–5836. Curran Associates, Inc., 2020. URL  
761 [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/](https://proceedings.neurips.cc/paper_files/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf)  
762 [3fe78a8acf5fda99de95303940a2420c-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf).

756 Yixuan Yu, Meng Li, Ruifeng He, et al. Mega: Parameter-efficient unlearning for large language  
757 models. *arXiv preprint arXiv:2312.08064*, 2023.  
758

759 Haibo Zhang, Toru Nakamura, Takamasa Isohara, and Kouichi Sakurai. A review on machine  
760 unlearning. *SN Computer Science*, 4(4):337, 2023.

761 Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catas-  
762 trophic collapse to effective unlearning. *First Conference on Language Modelling*, 2024. URL  
763 <https://openreview.net/pdf?id=MXLBXjQkmb>.  
764

765 Yuji Zhang, Sha Li, Cheng Qian, Jiateng Liu, Pengfei Yu, Chi Han, Yi R. Fung, Kathleen McKeown,  
766 ChengXiang Zhai, Manling Li, and Heng Ji. The law of knowledge overshadowing: Towards un-  
767 derstanding, predicting and preventing LLM hallucination. In Wanxiang Che, Joyce Nabende,  
768 Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Com-  
769 putational Linguistics: ACL 2025*, pp. 23340–23358, Vienna, Austria, July 2025. Association for  
770 Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1199.  
771 URL <https://aclanthology.org/2025.findings-acl.1199/>.  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809

## APPENDIX

## A DISCUSSION

A.1 DISCUSSION ON  $H$ 

We work in parameter space  $\mathbb{R}^p$ . Let  $H \succ 0$  denote the optimizer-induced metric; for Adam,  $H = W^\top W$  with

$$W = \text{diag}(1/\sqrt{\hat{v}} + \varepsilon), \quad (13)$$

where  $\hat{v}$  is Adam’s second-moment accumulator. For an optimizer with a (possibly time-varying) linear preconditioner  $P_t$  such that the step direction is  $d_t = -P_t g_t$ , set the metric to

$$H_t := P_t^\top P_t,$$

and use  $H_t$  consistently to build the retain basis  $U$  and the projectors  $P_{T_r}^{(H_t)}, P_\perp^{(H_t)}$ . Then the local retain invariance  $\Delta^{(1)} L_r = \langle \nabla L_r, \Delta \theta \rangle_{H_t} = 0$  is equivalent to  $\Delta \theta \in T_r^\perp$  under  $H_t$ , and all first-order safety statements carry the same.

## 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 gradients to form  $U$  and orthonormalize in whitened coordinates (Gram–Schmidt) so that  $U^\top H U = I_k$ . For any  $v \in \mathbb{R}^p$ , the projection  $P_\perp^{(H)} v$  lies in  $T_r^\perp$  and satisfies  $U^\top H P_\perp^{(H)} v = 0$ . Hence its  $H$  inner products with all retain-tangential directions vanish. Equivalently,  $P_\perp^{(H)}$  removes the tangential component along  $T_r$  and preserves only the  $H$  normal component, thereby eliminating retain–forget entanglement to first order while keeping the component that drives forgetting.

## A.3 SIGN-AWARE SELECTIVE PROJECTION

Not all retain-tangential components of  $\nabla L_f$  are harmful to  $L_r$ . In whitened coordinates  $\tilde{g}_f := W \nabla L_f$ ,  $\tilde{g}_r := W \nabla L_r$ , let  $a_i = \langle \tilde{g}_f, u_i \rangle$ ,  $b_i = \langle \tilde{g}_r, u_i \rangle$ . We keep only the *opposite-signed* retain-tangential components of  $\tilde{g}_f$  (which locally *decrease*  $L_r$ ), and discard same-signed (harmful) ones. We also cap their magnitude to avoid drifting within  $T_r$ :

$$\text{Keep } u_i \text{ if } a_i b_i < -\tau \quad (\tau \geq 0), \quad \left\| \sum_{i: a_i b_i < -\tau} a_i u_i \right\| \leq \kappa \| P_\perp^{(H)} \tilde{g}_f \| \quad (0 < \kappa \leq 1). \quad (14)$$

The resulting forget direction in whitened coordinates is  $\tilde{g}_f^{\text{sel}} = P_\perp^{(H)} \tilde{g}_f + \text{tan\_keep}$ , while the retain direction is  $\tilde{g}_r^{\text{nor}} = P_{T_r}^{(H)} \tilde{g}_r$ . Mapping back with  $W^{-1}$  yields the final gradient  $\nabla_\theta \leftarrow \gamma g_f^{\text{sel}} + \alpha g_r^{\text{nor}}$ . This preserves first-order safety (harmful tangential parts are removed) while not wasting helpful opposite-signed components.

**Sign-aware refinement.** With the sign-aware rule (Eq. equation 14), 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  $-\rho \sum_{i \in \mathcal{K}} a_i b_i \leq -\rho \tau \sum_{i \in \mathcal{K}} |a_i| |b_i| \leq 0$ , so Proposition 3.3 strengthens to

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

The cap in Eq. equation 14 further bounds the tangential energy, preventing drift within  $T_r$ .

#### 864 A.4 DISCUSSION ON PROJECTION-BASED UNLEARNING METHODS

865  
866  
867 In this section we provide a more detailed comparison between GU and prior projection-based un-  
868 learning methods, and we explain why we do not include UNSC/PGU/SEMUs ablations in our LLM  
869 experiments. UNSC (Chen et al., 2024) and PGU (Hoang et al., 2024) were designed for small- to  
870 medium-scale image classifiers (e.g., ResNet/ViT) with a small, fixed label space, where one can  
871 compute per-class activation subspaces or full-dataset Gram matrices and use them to define a null  
872 space that protects retain classes. SEMU (Sendra et al., 2025) similarly operates in a supervised  
873 setting, but constructs a low-rank “forget subspace” via SVD of forget-set gradients and parameter-  
874 izes unlearning updates inside this subspace. In contrast, our setting is large-scale LLM unlearning  
875 for offline SFT and preference-tuning, where outputs are open-vocabulary sequences, datasets are  
876 used post hoc for unlearning, and we must jointly optimize forgetting and retain performance in a  
877 multi-stage RLHF-style pipeline. The structural and data assumptions behind UNSC/PGU/SEMUs  
878 simply do not match this regime.

878 Conceptually, all these methods use “projection”, but they operate on different geometric ob-  
879 jects. UNSC and PGU define protected directions through representation or dataset statistics  
880 (class-conditional activation subspaces, retain-only Gram matrices), hoping that preserving these  
881 proxies approximately preserves retain behavior. SEMU, on the other hand, focuses on a forget-  
882 subspace derived solely from  $D_f$  and does not impose an explicit retain-side constraint; reten-  
883 tion is left to low rank and small step sizes. GU instead defines the retain geometry directly  
884 in parameter space as the span of retain gradients under the optimizer-induced SPD metric  $H$ ,  
885  $T_r(\theta) = \text{span} \nabla_{\theta} \ell_r(x_r; \theta) : x_r \in D_r$ . We prove that local retain invariance is equivalent to orthog-  
886 onality to  $T_r(\theta)$  under  $H$ , and that within the retain-orthogonal set the GU update is the steepest  
887 descent direction for the unlearning objective. Thus, in GU the retain-gradient subspace is the pri-  
888 mary geometric object, tightly coupled to the optimizer’s geometry, rather than an indirect proxy  
889 constructed from activations, Grams, or low-rank parameterizations.

890 A second key distinction is scalability. UNSC and PGU require computing and factorizing per-class  
891 activation covariance matrices or full-dataset Grams for each layer, and SEMU requires per-layer  
892 SVDs of gradient or weight-sized matrices. For LLMs with hidden dimension  $d \approx 4\text{k}-8\text{k}$  and  
893 hundreds of layers, these are  $d \times d$  objects whose storage and SVD cost are prohibitive, especially  
894 when unlearning must be performed repeatedly and on top of existing SFT. Any “lightweight” ap-  
895 proximation (e.g., collapsing outputs into a few pseudo-classes, dropping most layers, or heavily  
896 subsampling statistics) would deviate substantially from the original algorithms, making negative  
897 results difficult to interpret. GU is designed to avoid such  $d^2$ -scale computations: we never form  
898 activation or Gram matrices and never run layer-wise SVDs. Instead, we maintain a low-rank retain  
899 basis  $B \in \mathbb{R}^{d \times k}$  in parameter space using streaming updates from retain gradients, and project for-  
900 get gradients via vector-level operations  $g_f^\perp = g_f - BB_H^\top g_f$  with  $O(kd)$  cost under the optimizer  
901 metric  $H$ . This geometric layer introduces only a small overhead on top of existing unlearning  
902 algorithms (SimNPO, SatImp, WGA, NPO), which makes GU practically deployable at LLM scale.

902 For these reasons, we do not include UNSC/PGU/SEMUs as baselines in our LLM experiments.  
903 A faithful implementation of UNSC/PGU at LLM scale is essentially infeasible, while heavily ap-  
904 proximated variants would no longer reflect the original methods and would not yield clean scientific  
905 conclusions. SEMU’s forget-only formulation is, in principle, portable but optimizes a different ob-  
906 jective, maximizing forgetting along a low-rank  $D_f$ -dominated subspace without enforcing retain  
907 invariance, which is the central object of study in GU on TOFU/MUSE/WMDP. Instead, we focus  
908 on comparisons against strong and widely-used LLM-suitable unlearning baselines and on ablations  
909 that directly probe GU’s geometry (choice of metric, subspace rank), which more faithfully answer  
910 the question of whether geometric disentanglement improves the Pareto trade-off between forgetting  
911 and retention in the LLM alignment regime we target.

## 912 B ALGORITHM DETAILS

913  
914  
915 Here, we introduce the detailed practical implementation of Geometric-Disentanglement Unlearning  
916 (GU) in Algorithm 1.  
917

**Algorithm 1** Geometric-disentanglement Unlearning GU

**Require:** At step  $t$ , Parameters  $\theta_t$ , optimizer-induced metric  $H_t$  (whitener  $W_t$ ), reference model  $\theta_{\text{ref}}$ , batches  $B_f, B_r$ , weights  $\gamma, \alpha$

- 1: Compute forget loss  $L_f(\theta_t; B_f)$  and retain loss  $L_r(\theta_t; B_r, \theta_{\text{ref}})$ , 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  $L_r^{\text{KL}}$  and gradient  $g_r$ , whiten  $\tilde{g}_r = W_t g_r$  and update the retain basis  $U_t$  to approximate  $T_r(\theta_t)$ .
- 3: Recover forget gradient via  $g_f = (g_{\text{tot}} - \alpha g_r)/\gamma$  and whiten  $\tilde{g}_f = W_t g_f$
- 4: Decompose  $\tilde{g}_f$  w.r.t.  $U_t$  under  $H_t$ : obtain retain-orthogonal  $\tilde{g}_f^{\perp}$  and a sign-selective, norm-capped tangent part  $\tilde{g}_f^{\text{tan,keep}}$ ; similarly get  $\tilde{g}_r^{\text{tan}}$
- 5: Form whitened GU direction  $\tilde{g}_{\text{GU}} = \gamma(\tilde{g}_f^{\perp} + \tilde{g}_f^{\text{tan,keep}}) + \alpha \tilde{g}_r^{\text{tan}}$ , map back  $g_{\text{GU}} = W_t^{-1} \tilde{g}_{\text{GU}}$ , and let the base optimizer step with gradient  $g_{\text{GU}}$  to obtain  $\theta_{t+1}$

**Ensure:** Updated parameters  $\theta_{t+1}$

**Which parameters are projected.** 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.

**Metric  $H$  used by the projectors.** We work in coordinates whitened by the Adam preconditioner. For each selected parameter tensor  $p$ , let  $v_p$  denote Adam’s second moment estimate; then

$$W_p = \text{diag}\left(\frac{1}{\sqrt{v_p + \varepsilon}}\right), \quad H_p = W_p^{\top} W_p \quad (\text{diagonal}).$$

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 whitened coordinates  $\tilde{g} = W_p g$ . Thus,  $H$  is approximated by the Adam diagonal Fisher/Gauss–Newton surrogate already maintained during training.

**How we find  $P_{T_r}^{(H)}$  and  $P_{\perp}^{(H)}$ .** Per selected tensor  $p$ , we maintain a small basis  $U_p = \{u_{p,1}, \dots, u_{p,m}\}$  (with  $m \leq k$ ) that spans the retain tangent subspace  $T_r$  under  $H_p$ . 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  $g_{r,p} = \partial \mathcal{L}_r / \partial p$ , whiten  $\tilde{g}_{r,p} = W_p g_{r,p}$ , and run Gram–Schmidt against the current  $U_p$  in float32; if the relative residual  $\|\text{res}\| / \|g_{r,p}\|$  exceeds `residual_keep_thresh` and  $|U_p| < k$ , we append the normalized residual (stored in fp16). In whitened coordinates the projectors are

$$P_{T_r}^{(H)}(p) = U_p (U_p^{\top} U_p)^{-1} U_p^{\top}, \quad P_{\perp}^{(H)}(p) = I - P_{T_r}^{(H)}(p),$$

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, UNDIAl, SimNPO, CEU, WGA, SaT-IMP, or plain NLL). After the normal backward pass,  $g_{\text{tot},p} = \gamma g_{f,p} + \alpha g_{r,p}$  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) whiten:  $\tilde{g}_{f,p} = W_p g_{f,p}$ ,  $\tilde{g}_{r,p} = W_p g_{r,p}$ ; (iv) project:

$$\tilde{g}_{f,p}^{\text{safe}} = P_{\perp}^{(H)}(p) \tilde{g}_{f,p}, \quad \tilde{g}_{r,p}^{\text{tan}} = P_{T_r}^{(H)}(p) \tilde{g}_{r,p},$$

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

$$g_p^* = \gamma W_p^{-1} \tilde{g}_{f,p}^{\text{safe}} + \alpha W_p^{-1} \tilde{g}_{r,p}^{\text{tan}}.$$

The base optimizer (Adam) takes the step with its usual learning rate.

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

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

with  $\kappa =$  (default 0.5). We also use a sign threshold  $\tau =$  (default 0) and only keep tangential components where forget and retain gradients have *opposite* signs along the same basis vector ( $\langle \tilde{g}_{f,p}, u_{p,j} \rangle \cdot \langle \tilde{g}_{r,p}, u_{p,j} \rangle < -\tau$ ). The pair  $(\kappa, \tau)$  acts as a stable trust-region controller; the global step size remains the optimizer’s learning rate. We list  $\kappa$  and  $\tau$  in the hyperparameter table of the appendix.

## C PROOFS

### C.1 PROOF OF PROPOSITION 3.1

Formally, we reframe our Proposition 3.1 as Prop 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 (Euclidean) gradient  $g(x_r) := \nabla_\theta \ell_r(x_r; \theta) \in \mathbb{R}^p$ . Define the retain gradient subspace and its  $H$ -orthogonal complement*

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

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

$$L_r^S(\theta) := \sum_{i=1}^m \ell_r(x_{r_i}; \theta), \quad \nabla_\theta L_r^S(\theta) = \sum_{i=1}^m g(x_{r_i}) \in T_r(\theta).$$

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

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

Equivalently, the first-order quantity  $\Delta^{(1)} L_r^S(\theta; \Delta\theta) := \langle \nabla_\theta L_r^S(\theta), \Delta\theta \rangle_H$  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,  $\nabla_\theta L_r^S(\theta) = \sum_{i=1}^m g(x_{r_i}) \in T_r(\theta)$ .

(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  $\nabla_\theta L_r^S(\theta) \in T_r(\theta)$  for every finite  $S$ , we obtain  $\langle \nabla_\theta L_r^S(\theta), \Delta\theta \rangle_H = 0$  for all finite  $S \subset D_r$ .

(ii) $\Rightarrow$ (i). Assume  $\langle \nabla_\theta L_r^S(\theta), \Delta\theta \rangle_H = 0$  for all finite  $S \subset D_r$ . Take a singleton set  $S = \{x_r\}$ . Then  $\nabla_\theta L_r^S(\theta) = g(x_r)$  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,

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

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

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

**Remark.** If one prefers to identify  $\Delta^{(1)} L_r^S$  with the true directional derivative, introduce the  $H$ -gradient  $\nabla_\theta^H L := H^{-1} \nabla_\theta L$  so that  $D L_r^S(\theta)[\Delta\theta] = \langle \nabla_\theta^H L_r^S(\theta), \Delta\theta \rangle_H$ . The proof above is unchanged because  $\text{span}\{g(x_r)\}$  and  $\text{span}\{\nabla_\theta^H \ell_r(x_r; \theta)\}$  have the same  $H$ -orthogonal complement.

1026 C.2 PROOF OF LEMMA 3.2  
1027

1028 *Proof.* Because  $\mathcal{C} \subseteq T_r^\perp$ , every  $\Delta\theta \in \mathcal{C}$  satisfies  $\langle g_r, \Delta\theta \rangle_H = 0$  (since  $g_r \in T_r$ ). Hence for  
1029  $\Delta\theta \in \mathcal{C}$ ,

$$1030 \langle g_f + \alpha g_r, \Delta\theta \rangle_H = \langle g_f, \Delta\theta \rangle_H.$$

1032 Thus minimizing the joint directional derivative over  $\mathcal{C}$  is equivalent to minimizing  $\langle g_f, \Delta\theta \rangle_H$  over  
1033  $\mathcal{C}$ .

1034 Now decompose  $g_f = P_{T_r}^{(H)} g_f + P_\perp^{(H)} g_f$  with  $H$ -orthogonal components, and note that  $\Delta\theta \in T_r^\perp$   
1035 implies  $\langle P_{T_r}^{(H)} g_f, \Delta\theta \rangle_H = 0$ , so

$$1036 \langle g_f, \Delta\theta \rangle_H = \langle P_\perp^{(H)} g_f, \Delta\theta \rangle_H.$$

1040 By Cauchy–Schwarz,  $\langle P_\perp^{(H)} g_f, \Delta\theta \rangle_H \geq -\|P_\perp^{(H)} g_f\|_H \|\Delta\theta\|_H \geq -\|P_\perp^{(H)} g_f\|_H$ , with equality  
1041 achieved at  $\Delta\theta = -P_\perp^{(H)} g_f / \|P_\perp^{(H)} g_f\|_H$  when  $P_\perp^{(H)} g_f \neq 0$ . If  $P_\perp^{(H)} g_f = 0$ , then  $\langle g_f, \Delta\theta \rangle_H = 0$   
1042 for all  $\Delta\theta \in \mathcal{C}$ , so every feasible unit vector is optimal. Therefore the stated  $\Delta\theta_f^*$  solves both  
1043 problems over  $\mathcal{C}$ .  $\square$   
1044

1045 C.3 PROOF OF PROPOSITION 3.3  
1046

1047 *Proof.* We proceed in three explicit steps.  
1048

1049 First, we refer to the projector’s properties. By construction,  $P_{T_r}^{(H)}$  is the  $H$ -orthogonal projector  
1050 onto  $T_r$  and  $P_\perp^{(H)} := I_p - P_{T_r}^{(H)}$  is the projector onto  $T_r^\perp$ . Both are  $H$ -self-adjoint and idempotent,  
1051 and they are  $H$ -orthogonal in the sense that  $\langle P_{T_r}^{(H)} u, P_\perp^{(H)} v \rangle_H = 0$  for all  $u, v$ . Moreover, since  
1052  $g_r \in T_r$ , we have  $P_{T_r}^{(H)} g_r = g_r$  and  $P_\perp^{(H)} g_r = 0$ .

1053 By definition of the  $H$ -gradient,  $\Delta^{(1)} L_r = \langle g_r, \Delta\theta \rangle_H$  for any direction  $\Delta\theta$ . Substitute the split  
1054 step:  
1055

$$1056 \Delta^{(1)} L_r = \left\langle g_r, -\rho(P_\perp^{(H)} g_f + \beta P_{T_r}^{(H)} g_r) \right\rangle_H = -\rho \underbrace{\langle g_r, P_\perp^{(H)} g_f \rangle_H}_{(a)} - \rho\beta \underbrace{\langle g_r, P_{T_r}^{(H)} g_r \rangle_H}_{(b)}.$$

1061 For term (a):  $g_r \in T_r$  and  $P_\perp^{(H)} g_f \in T_r^\perp$ , hence by  $H$ -orthogonality,  $\langle g_r, P_\perp^{(H)} g_f \rangle_H = 0$ . For term  
1062 (b): since  $P_{T_r}^{(H)} g_r = g_r$ , we get  $\langle g_r, P_{T_r}^{(H)} g_r \rangle_H = \langle g_r, g_r \rangle_H = \|g_r\|_H^2$ . Therefore,  
1063

$$1064 \Delta^{(1)} L_r = -\rho\beta \|g_r\|_H^2 \leq 0,$$

1065 with strict inequality when  $\beta > 0$  and  $g_r \neq 0$ .  $\square$   
1066  
1067  
1068  
1069

1070 **Remark.** The statement and proof assume  $H$ -geometry consistently: inner products  $\langle \cdot, \cdot \rangle_H$ ,  $H$ -  
1071 gradients  $g_f = \nabla^H L_f$ ,  $g_r = \nabla^H L_r$ , and  $H$ -orthogonal projectors  $P_{T_r}^{(H)}$ ,  $P_\perp^{(H)}$ . If one uses Eu-  
1072 clidean gradients with the  $H$ -inner product, replace them by  $H$ -gradients via  $\nabla^H L = H^{-1} \nabla L$  to  
1073 keep the directional derivative  $\Delta^{(1)} L = \langle \nabla^H L, \Delta\theta \rangle_H$  consistent.  
1074

1075 In practice  $T_r$  is estimated from a mini-batch, yielding  $\widehat{T}_r$  and corresponding projectors. Then  
1076  $\langle g_r, P_\perp^{(H)} g_f \rangle_H$  may be small but nonzero, with magnitude controlled by the principal angle between  
1077  $T_r$  and  $\widehat{T}_r$ . The proposition captures the ideal (population) geometry; engineering deviations are  
1078  $O(\sin \Theta(T_r, \widehat{T}_r))$ .  
1079

Following Proposition 3.3, we proof the Corollary 3.4:

1080 *Proof.* Define  $\phi(\tau) := L_r(\theta + \tau\Delta\theta)$  for  $\tau \in [0, 1]$ . By the fundamental theorem of calculus and the  
 1081 definition of the  $H$ -gradient,  
 1082

$$\begin{aligned} 1083 L_r(\theta + \Delta\theta) - L_r(\theta) &= \int_0^1 \phi'(\tau) d\tau = \int_0^1 \langle \nabla_{\theta}^H L_r(\theta + \tau\Delta\theta), \Delta\theta \rangle_H d\tau \\ 1084 &= \langle \nabla_{\theta}^H L_r(\theta), \Delta\theta \rangle_H + \int_0^1 \langle \nabla_{\theta}^H L_r(\theta + \tau\Delta\theta) - \nabla_{\theta}^H L_r(\theta), \Delta\theta \rangle_H d\tau. \end{aligned}$$

1085  
 1086  
 1087 Apply Cauchy–Schwarz and the  $H$ -Lipschitz assumption:

$$\begin{aligned} 1088 \int_0^1 \langle \nabla_{\theta}^H L_r(\theta + \tau\Delta\theta) - \nabla_{\theta}^H L_r(\theta), \Delta\theta \rangle_H d\tau &\leq \int_0^1 \|\nabla_{\theta}^H L_r(\theta + \tau\Delta\theta) - \nabla_{\theta}^H L_r(\theta)\|_H \|\Delta\theta\|_H d\tau \\ 1089 &\leq \int_0^1 L_r^{(H)} \tau \|\Delta\theta\|_H^2 d\tau \\ 1090 &= \frac{L_r^{(H)}}{2} \|\Delta\theta\|_H^2. \end{aligned}$$

1091 Hence

$$1092 L_r(\theta + \Delta\theta) \leq L_r(\theta) + \langle \nabla_{\theta}^H L_r(\theta), \Delta\theta \rangle_H + \frac{L_r^{(H)}}{2} \|\Delta\theta\|_H^2.$$

1093 By Proposition 3.3,  $\langle \nabla_{\theta}^H L_r(\theta), \Delta\theta \rangle_H = -\rho\beta\|g_r\|_H^2$ . Moreover,  $P_{T_r}^{(H)} g_r \in T_r$  and  $P_{\perp}^{(H)} g_f \in T_r^{\perp}$   
 1094 are  $H$ -orthogonal, so

$$1095 \|\Delta\theta\|_H^2 = \rho^2 \|P_{\perp}^{(H)} g_f + \beta P_{T_r}^{(H)} g_r\|_H^2 = \rho^2 (\|P_{\perp}^{(H)} g_f\|_H^2 + \beta^2 \|g_r\|_H^2).$$

1096 Substitute these two identities into the previous inequality to obtain the stated bound. The strict-  
 1097 descent condition follows by requiring the quadratic upper bound to be negative, which yields the  
 1098 explicit upper bound on  $\rho$ .  $\square$

#### 1099 C.4 PROOF OF PROPOSITION 3.5

1100 *Proof.* Expand using bilinearity and  $H$ -orthogonality between  $T_r$  and  $T_r^{\perp}$ :

$$\begin{aligned} 1101 \langle g_f + \alpha g_r, \Delta\theta \rangle_H &= -\rho \left( \langle g_f, P_{\perp}^{(H)} g_f \rangle_H + \beta \langle g_f, P_{T_r}^{(H)} g_r \rangle_H + \alpha \langle g_r, P_{\perp}^{(H)} g_f \rangle_H + \alpha\beta \langle g_r, P_{T_r}^{(H)} g_r \rangle_H \right) \\ 1102 &= -\rho \left( \|P_{\perp}^{(H)} g_f\|_H^2 + \beta \langle P_{T_r}^{(H)} g_f, g_r \rangle_H + \alpha\beta \|g_r\|_H^2 \right), \end{aligned}$$

1103 since  $\langle g_r, P_{\perp}^{(H)} g_f \rangle_H = 0$  and  $P_{T_r}^{(H)} g_r = g_r$ .  $\square$

1104 **Corollary C.2** (Sufficient conditions for nonpositivity). *Under the setting of Proposition 3.5, the  
 1105 following hold:*

1106 (a) (No-repair case) *If  $\beta = 0$ , then  $\Delta^{(1)} \mathcal{L}_{\text{joint}} = -\rho \|P_{\perp}^{(H)} g_f\|_H^2 \leq 0$ .*

1107 (b) (With repair, unconditional bound) *For any  $\alpha > 0$ , by Cauchy–Schwarz and  $2ab \leq a^2 + b^2$ ,*

$$1108 \Delta^{(1)} \mathcal{L}_{\text{joint}} \leq -\rho \left( \|P_{\perp}^{(H)} g_f\|_H^2 + \frac{\alpha\beta}{2} \|g_r\|_H^2 - \frac{\beta}{2\alpha} \|P_{T_r}^{(H)} g_f\|_H^2 \right).$$

1109 *In particular, if*

$$1110 \|P_{\perp}^{(H)} g_f\|_H^2 + \frac{\alpha\beta}{2} \|g_r\|_H^2 \geq \frac{\beta}{2\alpha} \|P_{T_r}^{(H)} g_f\|_H^2,$$

1111 *then  $\Delta^{(1)} \mathcal{L}_{\text{joint}} \leq 0$ .*

1112 (c) (With repair, simple verifiable condition) *A sufficient, scale-invariant condition is*

$$1113 \alpha \|g_r\|_H \geq \|P_{T_r}^{(H)} g_f\|_H, \quad \text{under which} \quad \Delta^{(1)} \mathcal{L}_{\text{joint}} \leq -\rho \|P_{\perp}^{(H)} g_f\|_H^2 \leq 0.$$

Table 3: 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

*Proof.* (a) is the  $\beta = 0$  specialization of equation 12. For (b), bound the cross term using  $|\langle P_{T_r}^{(H)} g_f, g_r \rangle_H| \leq \|P_{T_r}^{(H)} g_f\|_H \|g_r\|_H \leq \frac{1}{2} (\frac{1}{\alpha} \|P_{T_r}^{(H)} g_f\|_H^2 + \alpha \|g_r\|_H^2)$ , then apply equation 12. For (c), if  $\alpha \|g_r\|_H \geq \|P_{T_r}^{(H)} g_f\|_H$  then  $\alpha \beta \|g_r\|_H^2 \geq \beta \|P_{T_r}^{(H)} g_f\|_H \|g_r\|_H \geq \beta |\langle P_{T_r}^{(H)} g_f, g_r \rangle_H|$ , so the bracket in equation 12 is at least  $\|P_{\perp}^{(H)} g_f\|_H^2$ , yielding the claim.  $\square$

*Remark C.3* (About “sign-aware” variants). One can enforce  $\langle P_{T_r}^{(H)} g_f, g_r \rangle_H \leq 0$  by modifying the step with a *sign-aware tangential gate*, but this may forfeit the retain monotonicity of Proposition 3.3. The unconditional and fully rigorous statements above therefore avoid such gates and instead provide transparent sufficient conditions. If a sign-aware mechanism is used, its rule and its effect on  $L_r$  must be stated explicitly to maintain rigor.

**From first-order to actual descent of the joint objective.** Under the same  $H$ -smoothness assumption as in Corollary 3.4, we also have

$$\mathcal{L}_{\text{joint}}(\theta + \Delta\theta) \leq \mathcal{L}_{\text{joint}}(\theta) + \Delta^{(1)} \mathcal{L}_{\text{joint}} + \frac{L_f^{(H)} + \alpha L_r^{(H)}}{2} \|\Delta\theta\|_H^2,$$

so any of the sufficient conditions in Corollary C.2 combined with a small enough stepsize (as in Corollary 3.4) yields an *actual* one-step decrease of  $\mathcal{L}_{\text{joint}}$ .

## D EXTRA EXPERIMENTS

### D.1 EXPERIMENT SETTING DETAILS

### 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 3). 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 4 shows 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 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. We will add these statistics (and analogous ones for MUSE/WMDP in the appendix) to make this global picture explicit. A similar Pareto improvement Figure is shown in Figure 3

### D.3 RESULTS ANALYSIS IN MUSE AND WMDP BENCHMARKS

**MUSE-Books: the full triad holds (forgetting  $\downarrow$ , retention  $\uparrow$ , leakage  $\rightarrow 0$ ).** Across all objective families, adding GU either lowers ES,Un or keeps it at the floor, raises ROUGE on the retain split, and moves privacy leakage closer to zero. Representative gains include SimNPO (ES,Un

Table 4: 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

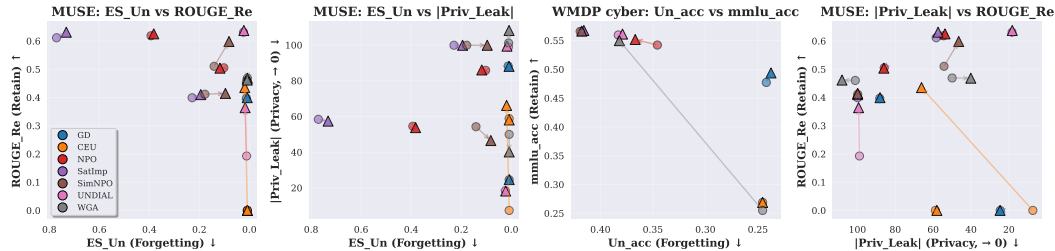


Figure 3: We visualize forgetting quality (ES Un: lower for better) against retained knowledge (ROUGE Re), privacy (Priv Leak) for eight unlearning baselines on MUSE and Un. acc. vs mmlu. acc. on WMDP. ROUGE Re and mmlu. acc. are metrics of higher for better. Priv Leak is a metric closer to 0 for better.

0.1407  $\rightarrow$  0.0813, ROUGE 0.5103  $\rightarrow$  0.5980, Priv. Leak  $-54.25 \rightarrow -46.49$ , SatImp (0.7710  $\rightarrow$  0.7321, 0.6114  $\rightarrow$  0.6310,  $-58.40 \rightarrow -57.39$ ), and NPO (0.3933  $\rightarrow$  0.3822, 0.6185  $\rightarrow$  0.6251,  $-54.49 \rightarrow -53.74$ ). For objectives already operating at minimal ES,Un (e.g., GD/CEU/WGA with 0.0079), GU maintains the floor while improving or preserving ROUGE and typically nudging leakage toward zero. This matches the mechanism: retain-null projection removes retain-tangent drift and, under the whitened metric, damps high-curvature entanglement, so forgetting efficacy is preserved while retained QA quality and privacy move in the desired directions.

**MUSE-News: retention gains are uniform; forgetting improves in the majority; leakage is largely neutral.** On News, GU consistently raises ROUGE on the retain split (e.g., GD 0.3971  $\rightarrow$  0.3992, CEU 0.0  $\rightarrow$  0.4349, SatImp 0.3991  $\rightarrow$  0.4100, SimNPO 0.4114  $\rightarrow$  0.4143, UNDIAL 0.1928  $\rightarrow$  0.3638, WGA 0.4602  $\rightarrow$  0.4615). Forgetting 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, UNDIAL) reflecting base-objective aggressiveness rather than instability. Privacy leakage on News is near-saturated for several methods (values around  $\pm 100$ ), so GU’s effect is mostly neutral; where the scale is moderate, GU tends to move leakage toward zero. Overall, the same geometric filter improves retention uniformly 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, UNDIAL 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, UNDIAL 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 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

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 truly removes the targeted factual memory rather than paraphrasing it, (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: Removing Parametric Memory Instead of Paraphrasing It** In the first forget case, both the base model and SimNPO continue to echo or paraphrase the synthetic fact, revealing residual parametric memory and incomplete privacy protection. In contrast, GU eliminates all identifiable traces of the original fact and reconstructs a plausible high-level motivation, demonstrating true forgetting rather than surface-level rewriting.

### Case Study 1: Erasing Parametric Memory Rather Than Copying It

**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 removes all identifiable traces of the synthetic knowledge and reconstructs a high-level and semantically plausible motivation (“personal experiences and professional interests in health, wellness, and nutrition”). This reflects true forgetting: GU erases the parametric encoding of the synthetic fact rather than its surface form, thereby preventing leakage 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 structural-pattern leakage. GU avoids regenerating any memorized templates and instead produces a high-level, semantically coherent description, showing selective forgetting without compromising topical understanding.

### Case Study 2: Unlearning Without Leaking Structural Patterns

**Subject:** Hina Ameen

**Question:** What was Hina Ameen’s maiden book?

1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349

**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 continue to rely on residual memory traces or memorized formatting patterns. This pattern-substitution behavior suggests that the forgetting process did not sever the underlying parametric pathways, but merely replaced the surface string. In contrast, GU abstracts the answer into a high-level, semantically appropriate description that contains no sensitive remnants (e.g., “A Guide to Geology: A Concise Introduction to the Earth’s Minerals”). This demonstrates that GU: avoids regenerating any structural patterns associated with the synthetic information, while still preserving a correct understanding of the topical domain (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 maintains alignment with the base model and ground truth, preserving thematic nuances, linguistic richness, and contextual reasoning without semantic degradation.

### 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 consistent 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 thematic 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 does not suffer from such over-forgetting. It continues to express fine-grained conceptual content rather than collapsing into vague or overly general statements, demonstrating that GU performs localized and selective forgetting without damaging the semantic structure of the retained knowledge.

**Retain Case: High-Fidelity Knowledge Retention Without Semantic Drift** In the full-retention scenario, GU faithfully preserves the original narrative structure, factual details, and reasoning quality, showing no unintended erasure or drift. SimNPO, however, hallucinates unrelated settings and elements, demonstrating that baseline unlearning can disrupt even non-targeted knowledge. GU achieves precise retention, preserving both accuracy and style.

#### 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 faithfully preserves the original narrative structure and semantic content—including cross-regional storytelling, the St. Petersburg setting, and the Chilean protagonist. It exhibits: no unintended erasure, no semantic drift, and intact reasoning and summarization abilities. 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 consistency, achieving “forgetting what must be forgotten and preserving what must be preserved,” thereby demonstrating high-precision, low-side-effect selective unlearning.

**Summary of qualitative trends.** Across all four cases, our qualitative analysis demonstrates that GU achieves precise, selective, and low-side-effect unlearning. In forget scenarios, GU fully removes the parametric memory of the synthetic facts without leaking structural patterns, while preserving topic coherence and linguistic fluency. In retain scenarios, GU avoids over-forgetting and maintains high-fidelity semantic content, narrative structure, and reasoning ability, whereas baseline methods frequently distort or hallucinate non-target knowledge. Together, these results highlight

1404 GU’s ability to “forget what must be forgotten and preserve what must be preserved”, achieving  
1405 reliable privacy protection without compromising the model’s utility.  
1406

1407 **F USAGE OF LLM**  
1408

1409 LLM is used to polish some of the writing of this paper.  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457