ROBUST LLM UNLEARNING AGAINST RELEARNING ATTACKS: THE MINOR COMPONENTS IN REPRESENTATIONS MATTER

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) unlearning aims to remove specific data influences from pre-trained model without costly retraining, addressing privacy, copyright, and safety concerns. However, recent studies reveal a critical vulnerability: unlearned models rapidly recover "forgotten" knowledge through relearning attacks. This fragility raises serious security concerns, especially for open-weight models. In this work, we investigate the fundamental mechanism underlying this fragility from a representation geometry perspective. We discover that existing unlearning methods predominantly optimize along dominant components, leaving minor components largely unchanged. Critically, during relearning attacks, the modifications in these dominant components are easily reversed, enabling rapid knowledge recovery, whereas minor components exhibit stronger resistance to such reversal. We further provide a theoretical analysis that explains both observations from the spectral structure of representations. Building on this insight, we propose **Minor Component Unlearning (MCU)**, a novel unlearning approach that explicitly targets minor components in representations. Extensive experiments on three datasets validate that, by concentrating unlearning effects in these inherently robust directions, our method achieves substantially improved resistance to relearning attacks.

1 INTRODUCTION

The rapid advancement of large language models (LLMs) has led to remarkable progress in various domains, from creative writing to code generation (Grattafiori et al., 2024). Meanwhile, open-weight models are being released at an increasing rate, with their capabilities lagging only six to twelve months behind closed-weight frontier models (Bhandari et al., 2025; Maslej et al., 2024). However, both open and closed models raise serious concerns about privacy violations, copyright infringement, and safety risks (Liu et al., 2025; Casper et al., 2025). When undesirable data influences are discovered post-deployment, retraining these massive models from scratch is often prohibitively expensive. This motivates the development of **LLM unlearning**, a post-training strategy that aims to remove specific data influences and suppress associated model capabilities without the need for complete retraining (Jang et al., 2023; Liu et al., 2025; Maini et al., 2024).

Despite the growing importance of LLM unlearning, several recent studies have identified a critical issue: **current unlearning methods lack robustness** (Łucki et al., 2025; Lynch et al., 2024; Hu et al., 2025; Deeb & Roger, 2024). Specifically, unlearned models exhibit a surprising susceptibility to quickly recovering "forgotten" knowledge through **relearning attacks** (Lynch et al., 2024; Hu et al., 2025). Even more concerning, fine-tuning on benign, unrelated downstream tasks can inadvertently undo the unlearning effects (Fan et al., 2025). For open-weight models, this lack of robustness poses severe security challenges: any downstream actor can easily reverse unlearning through minimal fine-tuning, undermining the intended protections (Casper et al., 2025; Rosati et al., 2024). Recent rigorous evaluations have revealed that state-of-the-art unlearning methods achieve recovery rates exceeding 88% after relearning attacks, demonstrating that they fail to truly remove knowledge from model weights (Deeb & Roger, 2024).

Although existing works have proposed various techniques to improve unlearning robustness—such as sharpness-aware minimization (SAM) (Fan et al., 2025) for smooth optimization and representation-

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

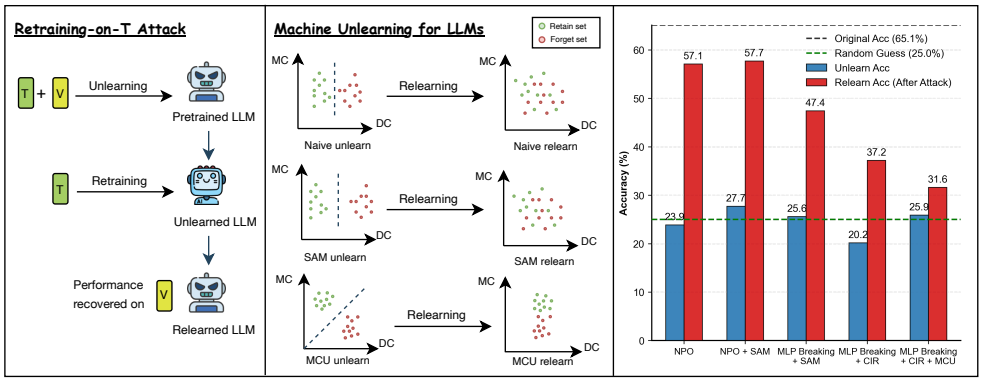


Figure 1: **Left:** Retraining-on-T (RTT) attack evaluation: the forget set is split into T and V; after unlearning on $T \cup V$, the attacker fine-tunes on T and measures recovery on V. **Middle:** Naive methods and SAM separate forget/retain representations mainly along dominant components (**DC**), which relearning easily reverses; MCU additionally separates them along minor components (**MC**), whose changes are largely preserved post-attack. **Right:** On WMDP-Cyber, MCU yields markedly lower post-attack accuracy while maintaining utility.

level interventions (Li et al., 2024; Sondej & Yang, 2025)—these methods remain largely empirical, and the fundamental mechanism underlying the susceptibility of LLM unlearning to relearning attacks remains poorly understood. We thus ask:

***(Q)** Why is LLM unlearning so fragile against relearning attacks?*

To address **(Q)**, we conduct a principled analysis of LLM unlearning through the lens of representation geometry. We discover that existing unlearning methods predominantly optimize along the dominant component directions, leaving minor components largely unchanged. Critically, when relearning attacks are applied, the modifications in these dominant components are easily reversed—with recovery rates significantly exceeding those of minor components—explaining why current methods are so vulnerable to such attacks. We further give a theoretical analysis that derives both phenomena from the spectral structure of representations, identifying the structural source of fragility.

Inspired by these findings, we propose **Minor Component Unlearning (MCU)**, a novel unlearning approach that explicitly targets the minor components of internal representations. By leveraging the observation that minor components are inherently more resistant to recovery during relearning, our method achieves substantially improved robustness against relearning attacks while maintaining model utility on unrelated tasks. We summarize our **contributions** below.

• We provide the first systematic analysis of LLM unlearning robustness from a representation geometry perspective, supported by a theoretical analysis. We identify a key mechanism underlying unlearning fragility: dominant components modified during unlearning are easily recovered by relearning attacks, whereas minor components exhibit significantly stronger resistance to recovery.

• Building on these insights, we propose MCU, a novel unlearning method that explicitly targets minor components of representations.

• We conduct extensive experiments on WMDP-Cyber, WMDP-Bio, and Years datasets, demonstrating that our method significantly reduces knowledge recovery after relearning attacks while preserving model utility, outperforming existing methods. Some experiment highlights on WMDP-Cyber dataset are showcased in Figure 1.

2 Preliminaries of LLM Unlearning

Problem Definition. LLM unlearning aims to erase or suppress undesirable knowledge within a pre-trained LLM while preserving its general performance (Liu et al., 2025). Formally, given a pre-trained LLM with parameters $\boldsymbol{\theta}_{\mathrm{o}}$ and a dataset partitioned into a **forget set** $\mathcal{D}_{\mathrm{f}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_f}$ containing data to be unlearned and a **retain set** $\mathcal{D}_{\mathrm{r}} = \{(\mathbf{x}_j, y_j)\}_{j=1}^{n_r}$ containing data the model

should still remember, unlearning seeks to obtain updated parameters $\boldsymbol{\theta}_{\mathrm{u}}$ such that the model "forgets" information in $\mathcal{D}_{\mathrm{f}}$ while maintaining performance on $\mathcal{D}_{\mathrm{r}}$. An ideal unlearning method should ensure that the mutual information between the unlearned model weights and the forget set approaches zero, meaning the removed knowledge is truly erased from the model rather than merely hidden (Deeb & Roger, 2024).

Unlearning Methods. Let $\pi_{\boldsymbol{\theta}}(x)$ denote the probability of text x under model parameters $\boldsymbol{\theta}$. **Gradient Ascent (GA)** (Jang et al., 2023) maximizes the cross-entropy loss on the forget set, while **Negative Preference Optimization (NPO)** (Zhang et al., 2024) adapts DPO (Rafailov et al., 2023) by treating $\mathcal{D}_{\mathrm{f}}$ as dispreferred responses:

$$\mathcal{L}_{\mathrm{GA}} = \mathop{\mathbb{E}}_{x \in \mathcal{D}_{\mathrm{f}}}[\log \pi_{\boldsymbol{\theta}}(x)], \tag{1}$$

$$\mathcal{L}_{\mathrm{NPO}} = -\frac{2}{\beta} \mathop{\mathbb{E}}_{x \in \mathcal{D}_{\mathrm{f}}} \left[\log \sigma \left(-\beta \log \frac{\pi_{\theta}(x)}{\pi_{\mathrm{ref}}(x)} \right) \right], \tag{2}$$

where $\pi_{\mathrm{ref}} = \pi_{\boldsymbol{\theta}_{\mathrm{o}}}$ and β is a temperature parameter. **Representation Misdirection for Unlearning (RMU)** (Li et al., 2024) perturbs internal hidden states toward a random control vector, while **MLP Breaking** (Sondej & Yang, 2025) drives MLP outputs toward orthogonality with their originals (motivated by factual knowledge being stored in MLP parameters (Nanda et al., 2023)):

$$\mathcal{L}_{\mathrm{RMU}} = \mathop{\mathbb{E}}_{x \in \mathcal{D}_{\mathrm{f}}} \left[\sum_{t \in x} \|\mathbf{h}(t) - c \cdot \mathbf{u}\|^2 \right], \tag{3}$$

$$\mathcal{L}_{\text{MLP Breaking}} = \mathop{\mathbb{E}}_{x \in \mathcal{D}_{\mathrm{f}}} \left[\sum_{t \in x} \mathrm{ReLU} \left(\frac{\langle \mathbf{h}(t), \mathbf{h}_{\mathrm{o}}(t) \rangle}{\|\mathbf{h}_{\mathrm{o}}(t)\|^2} \right) \right]. \tag{4}$$

where $\mathbf{h}(t)$ is the current internal representation of token t (hidden state for RMU, MLP output for MLP Breaking), $\mathbf{h}_{\mathrm{o}}(t)$ is its value under $\boldsymbol{\theta}_{\mathrm{o}}$, c is a scaling hyperparameter, and $\mathbf{u}$ is a random control vector.

3 Understanding Fragile LLM Unlearning

In this section, we investigate how unlearning and relearning affect an LLM's internal representations, and identify a structural cause of fragility: **unlearning predominantly modifies the dominant (high-variance) directions of internal representations, which are widely shared across samples and therefore easily reversed by relearning attacks.** Section 3.1 establishes this empirically through three observations, and Section 3.2 explains Observations 2 and 3 theoretically.

3.1 Empirical Observations on Representation Geometry

Setup. We extract MLP activations across all layers of Llama-3.1-8B on the forget set $\mathcal{D}_{\mathrm{f}}$ and apply PCA, yielding principal components $\{\mathbf{v}_1, \ldots, \mathbf{v}_d\}$ ordered by decreasing variance $\sigma_1^2 \geq \cdots \geq \sigma_d^2$. We then track how representations move along each PC during unlearning and relearning. Implementation details (modules used, sample sizes, layer aggregation) are deferred to Appendix F.

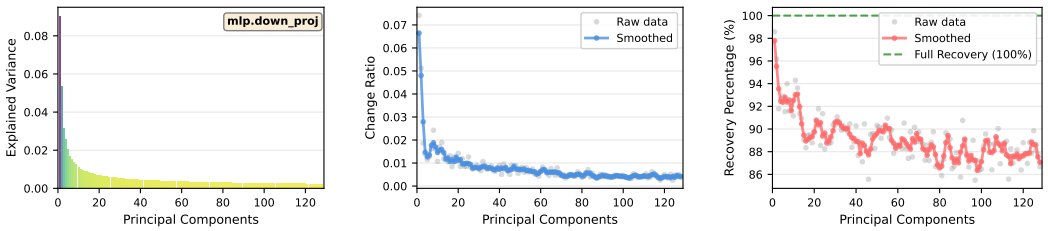
Observation 1: LLM Representations are Concentrated in Dominant Components. Figure 2a shows the explained-variance ratio across principal components: **the first few dominant components capture the overwhelming majority of the total variance**, while the minor components form a long tail of small but non-negligible contributions.

To quantify how unlearning and relearning affect each direction, we define two metrics for each principal component $\mathbf{v}_k$:

$$\text{Change Ratio}_k = \frac{|\langle \mathbf{h}_{\mathrm{u}} - \mathbf{h}_{\mathrm{o}}, \mathbf{v}_k \rangle|}{\sum_{j=1}^{d} |\langle \mathbf{h}_{\mathrm{u}} - \mathbf{h}_{\mathrm{o}}, \mathbf{v}_j \rangle|}, \tag{5}$$

$$\text{Recovery Ratio}_k = \frac{\langle \mathbf{h}_{\mathrm{u}} - \mathbf{h}_{\mathrm{r}}, \mathbf{v}_k \rangle}{\langle \mathbf{h}_{\mathrm{u}} - \mathbf{h}_{\mathrm{o}}, \mathbf{v}_k \rangle}, \tag{6}$$

where $\mathbf{h}_{\mathrm{o}}$, $\mathbf{h}_{\mathrm{u}}$, and $\mathbf{h}_{\mathrm{r}}$ denote representations before unlearning, after unlearning, and after a relearning attack, respectively; a Recovery Ratio near 1 indicates full reversal and near 0 indicates robust unlearning.



(a) Explained variance (b) Unlearning change (c) Relearning recovery

Figure 2: Principal component analysis of LLM representations during unlearning and relearning. (a) The first few dominant components capture the majority of variance in representations. (b) Unlearning predominantly modifies these dominant components, leaving minor components unchanged. (c) Relearning attacks preferentially recover the dominant components, making unlearning effects along these directions easily reversible.

Observation 2: Unlearning Predominantly Modifies Dominant Components. Figure 2b reports the Change Ratio (5) after applying GA: **unlearning induces disproportionately large changes along the leading PCs**, while minor components remain largely unchanged. The same pattern holds across unlearning losses (Appendix L).

Observation 3: Dominant Components are More Easily Recovered During Relearning. The concentration above would be benign if the changes were robust. Figure 2c reports the Recovery Ratio (6): **dominant components attain substantially higher recovery ratios (often $>90\%$) than minor components**, with the same pattern across unlearning losses.

3.2 Theoretical Analysis of Unlearning Fragility

The empirical patterns are not coincidental: they follow from the spectral structure of forget-set representations and the gradient geometry of unlearning/relearning losses. We formalize this through a linearized (NTK-style) analysis that yields the two theorems below; full derivations and assumptions are deferred to Appendix D.

Theorem 1 (Dominant-component concentration; explains Observation 2)**.** *After T unlearning steps,*

$$\mathbb{E}_{\mathcal{D}_f}\big[\langle \mathbf{h}_u - \mathbf{h}_o, \mathbf{v}_k\rangle^2\big] \;\propto\; \sigma_k^2 \;+\; O(\tau^2), \tag{7}$$

with $\tau^2 \ll \sigma_1^2$ a small noise term. The change-ratio (5) *therefore mirrors the explained-variance profile of Figure 2a: the unlearning update is channeled through directions that account for most of the representation variance.*

Theorem 2 (Dominant-component recoverability; explains Observation 3)**.** *Because the relearning distribution $\mathcal{D}_r$ shares its dominant eigenstructure with $\mathbf{\Sigma}$ (standard threat model), applying the same NTK linearization to the relearning objective gives, for some $c>0$ and T_r relearning steps,*

$$\text{Recovery Ratio}_k \;\approx\; 1-\exp\big(-c\,\sigma_k^2\,T_r\big). \tag{8}$$

Dominant components saturate within a few attack steps, while minor components require $O(1/\sigma_k^2)$ steps and remain effectively unrecovered under any bounded budget. A complementary argument (Appendix D.5) further shows that, even as $T_r\to\infty$, the relearning gradients along minor components average out across the batch, so these directions cannot be reliably reconstructed by the attacker.

Together, Theorems 1 and 2 show that the unlearning effect is concentrated in exactly the directions the attacker can recover most cheaply—a structural source of fragility. Redirecting unlearning into the minor-component subspace inverts the scaling (8) and works *against* the attacker; we develop this idea in Section 4.

4 Methodology

Building on Section 3, we propose **Minor Component Unlearning (MCU)**, which explicitly redirects representation-based unlearning losses (e.g., RMU (Li et al., 2024), MLP Breaking (Sondej & Yang, 2025)) toward the minor-component subspace, in line with the structural fragility characterized by Theorems 1 and 2.

4.1 Motivation: Targeting Minor Components

Theorems 1 and 2 together imply that an unlearning update concentrated in dominant directions is exactly the configuration the attacker can undo most cheaply. This motivates a natural question: **can we design an unlearning method that confines its effect to minor components, thereby inheriting their resistance to relearning?** To achieve this, we propose to remove the dominant components from both the current and target representations before computing the unlearning loss. By removing the high-variance directions, the resulting loss gradients are constrained to operate primarily within the minor component subspace. This ensures that unlearning-induced changes occur in directions that are inherently more difficult to reverse.

4.2 Principal Component Extraction

Before unlearning, we extract the principal components from the original model's representations on the forget set $\mathcal{D}_\mathrm{f}$. For each trainable parameter, we collect internal representations (either hidden states for RMU or MLP outputs for MLP Breaking) across all tokens in the forget set. Let $\mathbf{H} \in \mathbb{R}^{N \times d}$ denote the matrix of collected representations, where N is the total number of tokens and d is the hidden dimension.

We first center the representations by subtracting the mean:

$$\bar{\mathbf{h}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{h}_i, \quad \tilde{\mathbf{H}} = \mathbf{H} - \mathbf{1}\bar{\mathbf{h}}^\top, \tag{9}$$

where $\mathbf{1}$ is the all-ones vector. We then compute the top $K-1$ principal components $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{K-1}\}$ via singular value decomposition (SVD) or power iteration (Halko et al., 2011):

$$\tilde{\mathbf{H}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \quad \mathbf{v}_k = \mathbf{V}_{:,k}. \tag{10}$$

In practice, we use randomized SVD for computational efficiency, which computes an approximate low-rank decomposition with $O(NK^2)$ complexity instead of $O(Nd^2)$ for full SVD.

4.3 Minor Component Projection

Given the extracted principal components, we define the projection operator that removes a rank-K subspace from any representation $\mathbf{h} \in \mathbb{R}^d$, where K is a hyperparameter and $\langle \cdot, \cdot \rangle$ denotes the inner product. We treat the mean representation $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{h}_i$ as a special "0th" principal component. Concretely, the removed subspace consists of the mean direction together with the top $K-1$ centered principal components:

$$\mathcal{P}_\perp(\mathbf{h}) = \mathbf{h} - \sum_{k=0}^{K-1} \langle \mathbf{h}, \mathbf{v}_k \rangle \mathbf{v}_k, \tag{11}$$

where $\mathbf{v}_0 = \boldsymbol{\mu}/\|\boldsymbol{\mu}\|$ and $\{\mathbf{v}_1, \ldots, \mathbf{v}_{K-1}\}$ are the top centered principal components. The projected vector $\mathcal{P}_\perp(\mathbf{h})$ lies in the orthogonal complement of this subspace, *i.e.*, the minor component subspace. Empirically, including the mean yields consistently better unlearning robustness; we provide ablation results in Section 5.3.

4.4 RMU-MCU

RMU (Li et al., 2024) aims to steer the hidden representations towards a random target vector, disrupting the model's ability to produce forget-set-related outputs. The original RMU loss is given in Equation 3.

Our Minor Component Unlearning variant, **RMU-MCU**, applies the minor component projection to the current representation before computing the loss:

$$\mathcal{L}_{\text{RMU-MCU}} = \mathop{\mathbb{E}}_{x \in \mathcal{D}_\mathrm{f}} \left[\sum_{t \in x} \| \mathcal{P}_\perp(\mathbf{h}(t)) - c \cdot \mathbf{u} \|^2 \right]. \tag{12}$$

Intuition. Consider the gradient of (12) *with respect to the hidden representation* $\mathbf{h}(t)$:

$$\frac{\partial \mathcal{L}_{\text{RMU-MCU}}}{\partial \mathbf{h}(t)} \propto \mathcal{P}_{\perp}(\mathbf{h}(t)) - c \cdot \mathbf{u}. \tag{13}$$

The input-dependent part $\mathcal{P}_{\perp}(\mathbf{h}(t))$ lies in the minor component subspace, so MCU injects unlearning pressure exclusively along these robust directions; in contrast, standard RMU's gradient $\mathbf{h}(t) - c \cdot \mathbf{u}$ has large components along principal directions, leading to easily reversible changes. The corresponding parameter-space update is this signal pulled back through the post-$\mathbf{h}$ Jacobian: while the update need not be strictly confined to the minor subspace, the dominant-direction contribution to $\mathbb{E}[\mathbf{g}_u \mathbf{g}_u^{\top}]$—the term that drives Theorems 1 and 2—is removed.

4.5 MLP-Breaking-MCU

The original MLP Breaking loss aims to make the current MLP outputs orthogonal to the original outputs, as given in Equation 4.

Our variant, **MLP-Breaking-MCU**, applies minor component projection before computing the loss:

$$\mathcal{L}_{\text{MLP-Breaking-MCU}} = \mathop{\mathbb{E}}_{x \in \mathcal{D}_{\text{f}}} \left[\sum_{t \in x} \text{ReLU}\left(\frac{\langle \mathcal{P}_{\perp}(\mathbf{h}(t)), \mathcal{P}_{\perp}(\mathbf{h}_{\text{o}}(t)) \rangle}{\|\mathcal{P}_{\perp}(\mathbf{h}_{\text{o}}(t))\|^2} \right) \right]. \tag{14}$$

Intuition. When the ReLU is active, the gradient of (14) *with respect to* $\mathbf{h}(t)$ is:

$$\frac{\partial \mathcal{L}_{\text{MLP-Breaking-MCU}}}{\partial \mathbf{h}(t)} \propto \frac{\mathcal{P}_{\perp}(\mathbf{h}_{\text{o}}(t))}{\|\mathcal{P}_{\perp}(\mathbf{h}_{\text{o}}(t))\|^2}. \tag{15}$$

This vector lies in the minor component subspace by construction, since $\mathcal{P}_{\perp}(\mathbf{h}_{\text{o}}(t))$ is orthogonal to the principal directions. As in Equation (12), the parameter update is the pull-back of this hidden-state signal through the post-$\mathbf{h}$ Jacobian; what is guaranteed is that the dominant-direction component of the residual covariance vanishes, removing the σ_k^2-scaling source of fragility identified in Theorems 1 and 2.

5 Experiments

5.1 Experiment Setups

Datasets. We use three forget sets: **WMDP-Cyber** and **WMDP-Bio** from the WMDP-Deduped benchmark (Deeb & Roger, 2024) (deduplicated subsets of WMDP (Li et al., 2024), further filtered following Sondej & Yang (2025)), and **Years** (Deeb & Roger, 2024) (20th-century events with their dates). Retain sets are domain-matched subsets of the FineFineWeb corpus (M-A-P et al., 2024). Full splits and preprocessing are in Appendix G.

Evaluation. Following Deeb & Roger (2024), we partition the forget set into disjoint T (80%) and V (20%) with minimal mutual information, unlearn on $T \cup V$, and measure post-attack recovery on V after the **RTT** attack (fine-tuning the unlearned model on T). We report five metrics: **MMLU** (general knowledge $\uparrow$), **WikiText** loss (Merity et al., 2016) ($\downarrow$), **Forget** accuracy ($\downarrow$), **Relearn** accuracy after RTT ($\downarrow$), and the relearning gap $\mathbf{\Delta}$ = Relearn − Forget ($\downarrow$). All experiments use Llama-3.1-8B (Grattafiori et al., 2024) unless noted.

Baselines. We evaluate three base unlearning losses (NPO, RMU, and MLP Breaking) combined with two robustness-oriented techniques: Sharpness-Aware Minimization (SAM) (Fan et al., 2025) and Collapse of Irrelevant Representations (CIR) (Sondej & Yang, 2025). CIR is complementary to MCU: it operates at the *gradient level* by projecting out dominant components from $\nabla_\theta \mathcal{L}_u$ before each update, whereas MCU operates at the *loss level* by reshaping which directions the loss penalizes. We therefore evaluate MCU both standalone and on top of CIR. We discuss the spectral interpretation of CIR and the additional contributions of our work in Appendix E.

Table 1: Main results on WMDP-Cyber, WMDP-Bio, and Years. For Relearn and Δ, **bold** = best, underline = second-best per dataset.

Dataset	Method	MMLU (↑)	WikiText (↓)	Forget (↓)	Relearn (↓)	Δ (↓)
WMDP-Cyber	Original model	65.1	1.000	57.6	-	-
	RMU	52.8	1.207	28.7	54.6	26.0
	RMU + MCU	49.3	1.202	28.8	53.9	25.1
	RMU + CIR	64.1	1.006	28.0	50.5	22.5
	RMU + CIR + MCU	64.7	1.006	31.7	43.3	11.6
	MLP Breaking	61.8	1.132	27.3	55.8	28.4
	MLP Breaking + MCU	60.4	1.105	26.0	52.0	26.0
	MLP Breaking + CIR	65.0	1.001	20.2	37.2	17.0
	MLP Breaking + CIR + MCU	64.7	1.001	25.9	**31.6**	**5.7**
WMDP-Bio	Original model	65.1	1.000	64.0	-	-
	RMU	45.5	1.300	26.7	49.8	23.1
	RMU + MCU	46.5	1.205	28.0	48.7	20.7
	RMU + CIR	63.7	1.010	27.9	47.6	19.7
	RMU + CIR + MCU	63.5	1.010	32.2	39.5	7.3
	MLP Breaking	58.7	1.364	29.9	58.8	28.9
	MLP Breaking + MCU	60.3	1.357	29.1	57.5	28.5
	MLP Breaking + CIR	64.8	1.001	22.1	29.7	7.6
	MLP Breaking + CIR + MCU	64.5	1.001	22.8	**26.4**	**3.6**
Years	Original model	65.1	1.000	68.4	-	-
	RMU	56.8	1.204	33.0	64.3	31.4
	RMU + MCU	52.0	1.193	30.8	54.9	24.1
	RMU + CIR	57.1	1.102	30.7	37.4	6.7
	RMU + CIR + MCU	57.2	1.101	31.7	33.7	**2.0**
	MLP Breaking	60.8	1.239	27.0	51.5	24.5
	MLP Breaking + MCU	61.5	1.214	29.0	49.5	20.6
	MLP Breaking + CIR	64.6	1.010	32.7	39.4	6.7
	MLP Breaking + CIR + MCU	63.8	1.010	25.9	**30.9**	5.0

5.2 Main Results

Table 1 summarizes our main results across all three datasets. We report a compact subset of representative configurations to keep the table readable; remaining configurations are provided in Table 5 of Appendix H. We additionally evaluate an adaptive representation-based attack in Appendix J.

Adding MCU improves relearning robustness. Table 1 pairs each base loss with the same loss plus MCU. Across WMDP-Cyber, WMDP-Bio, and Years, RMU + MCU and MLP Breaking + MCU are more robust to relearning than the base loss: both reach lower post-attack Relearn accuracy and a smaller relearning gap Δ, and the two metrics move together on every dataset.

MCU and CIR provide complementary benefits. Relative to either base loss alone, CIR improves utility retention and improves the relearning robustness, yet CIR alone still leaves room for knowledge recovery under relearning attacks. Adding MCU on top of CIR achieves substantially lower post relearning accuracy and (Δ) than all other configurations, demonstrating that the two techniques address different aspects of the unlearning robustness problem. This improvement is particularly pronounced when MCU is applied on top of MLP Breaking + CIR, where we observe the lowest Δ values across all three datasets. Notably, MCU provides these robustness gains without compromising model utility—the MMLU scores of MCU variants remain comparable to or better than their non-MCU counterparts, while WikiText perplexity stays near the original model baseline.

SAM provides limited robustness improvements. While SAM has been proposed to improve unlearning robustness through smoother loss landscapes (Fan et al., 2025), the full comparison in Table 5 (Appendix H) shows that its effectiveness varies across settings. In some cases, SAM slightly reduces the relearning gap, but the improvements are inconsistent and often come with utility

Table 2: Rank-matched controls for MLP Breaking + CIR. **Bold**: lowest Relearn and Δ.

Dataset	Removed subspace	K	MMLU (↑)	WikiText (↓)	Forget (↓)	Relearn (↓)	Δ (↓)
WMDP-Cyber	Top (MCU)	32	64.7	1.001	25.9	**31.6**	**5.7**
	Middle	32	65.1	1.001	19.6	36.9	17.4
	Random	32	65.2	1.001	19.9	37.6	17.7
WMDP-Bio	Top (MCU)	8	64.5	1.001	22.8	**26.4**	**3.6**
	Middle	8	64.6	1.001	22.5	29.3	6.7
	Random	8	64.8	1.001	21.6	29.7	8.1

trade-offs. In contrast, MCU provides more reliable and substantial robustness gains, suggesting that operating on the representation geometry is more effective than loss landscape smoothing for preventing knowledge recovery.

Generality across model families. The results in Table 1 are reported on Llama-3.1-8B for direct comparability with prior work (Deeb & Roger, 2024; Sondej & Yang, 2025). To verify that the benefits of MCU are not specific to a single model, we additionally evaluate MCU on **Gemma2-9B** and **Qwen3-8B** across all three datasets; the full results are provided in Appendix K (Table 8). On both architectures, adding MCU consistently reduces the post-attack relearning gap Δ over the strong MLP Breaking + CIR baseline while preserving MMLU, confirming that the improvements transfer across model families.

5.3 Ablation Studies

Robustness Comes from Removing Dominant Directions. To strengthen the causal evidence beyond the change and recovery analysis in Section 3, we compare three removed subspaces of the same rank K, with all other training and evaluation settings fixed. The base method is MLP Breaking + CIR. **(1) Top (MCU):** the mean direction plus the top $K-1$ centered principal components; **(2) Middle:** principal components $K+1$ through $2K$, without the mean; **(3) Random:** K random orthonormal directions. On WMDP-Cyber we use $K=32$ and on WMDP-Bio we use $K=8$, the dataset-specific settings used in Table 1. This design tests whether robustness comes merely from removing any K-dimensional subspace, or specifically from avoiding the dominant directions that are easily recovered during relearning. Table 2 shows that top-PC removal gives the lowest Relearn and Δ on both datasets, while MMLU and WikiText remain nearly identical across strategies. These rank-matched interventions support the dominant/minor spectral distinction.

Changes Across Principal Components. To validate that our method successfully redirects unlearning effects toward minor components as intended, we analyze the distribution of representation changes across principal components after unlearning. Specifically, we extract the principal components from the original model's representations on the forget set (as described in Section 4), then measure how much each component is modified during unlearning for both baseline and our MCU variants. For each principal component $\mathbf{v}_k$, we compute the change ratio as defined in Equation 5. Figure 3 presents the distribution of changes across principal component bins for baseline (MLP Breaking + CIR) compared to our MCU variants. A clear pattern emerges: **baseline concentrate the majority of representation changes in the early bins, corresponding to the dominant principal components that encode shared structure and are easily recovered during relearning**. In contrast, **our MCU variants shift the distribution toward later bins, indicating that changes are redistributed to the minor components that store sample-specific information**. This shift aligns precisely with our design objective—by projecting out the rank-K subspace (the mean direction together with the top $K-1$ centered principal components) before computing the unlearning loss, MCU suppresses modifications to the dominant subspace and redirects optimization pressure toward the minor component subspace. The redistribution of unlearning effects provides a mechanistic explanation for the improved robustness observed in Table 1. As demonstrated in our

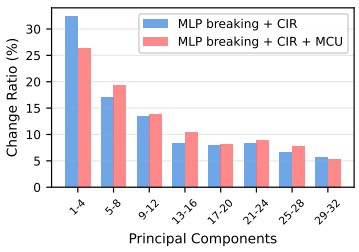


Figure 3: PC-bin change distribution. Baseline concentrates changes in dominant bins; MCU shifts mass toward minor bins.

Table 4: Sensitivity to K. Highlighted rows are the settings in Table 1.

WMDP-Cyber						WMDP-Bio					
K	MMLU↑	Wiki↓	Forget↓	Relearn↓	$\Delta\downarrow$	K	MMLU↑	Wiki↓	Forget↓	Relearn↓	$\Delta\downarrow$
2	64.9	1.001	26.1	32.4	6.2	2	64.6	1.001	20.3	36.7	16.4
4	64.9	1.001	25.3	33.3	8.0	4	64.7	1.001	21.4	24.3	2.9
8	64.4	1.002	24.8	32.3	7.5	8	64.5	1.001	22.8	26.4	3.6
16	64.7	1.002	24.9	31.3	6.4	16	64.6	1.001	24.8	25.0	0.2
32	64.7	1.001	25.9	31.6	5.7	32	64.7	1.001	25.6	24.2	−1.4

analysis (Section 3), modifications to dominant components can be easily reversed during relearning because these directions capture cross-sample regularities that the model naturally recovers when exposed to similar data.

Ablation Study on Component Projection Strategies. We ablate several ways of constructing the projection subspace, varying whether we explicitly handle the mean direction and whether we use centered PCA or SVD-style decomposition: **(1) MCU (Ours)**: compute the mean as the 0th component, then apply PCA on centered representations; **(2) w/o mean**: standard PCA without treating mean as a special component; **(3) SVD-based**: SVD on uncentered representations, using top-K right singular vectors. Table 3 shows that explicitly accounting for the mean direction is critical for achieving robust unlearning. When the mean component is not treated separately ("w/o mean"), the method fails to reliably isolate the recoverable directions, leading to unpredictable performance—in some cases even worse than the baseline. Similarly, SVD on uncentered representations provides moderate improvements but remains less effective than centered PCA, likely because the dominant singular vectors conflate the mean shift with principal variation directions. These results suggest that the mean direction captures globally shared information that is particularly susceptible to recovery, and explicitly projecting it out enables more precise targeting of the robust subspace. Importantly, all variants maintain similar utility scores, indicating that the performance differences primarily reflect how well each strategy identifies and avoids the recoverable directions rather than fundamental trade-offs with model capability.

Table 3: Ablation on projection strategies. All methods use MLP Breaking + CIR as base. Fgt = Forget acc., Rlrn = Relearn acc. after RTT.

Dataset	Method	MMLU	Fgt	Rlrn	Δ
WMDP-Cyber	CIR (baseline)	65.0	20.2	37.2	17.0
	+ SVD-based	64.6	22.4	31.1	8.7
	+ MCU w/o mean	64.4	22.3	31.7	9.4
	+ MCU	64.7	25.9	31.6	**5.7**
WMDP-Bio	CIR (baseline)	64.8	22.1	29.7	7.6
	+ SVD-based	64.5	22.7	29.2	6.5
	+ MCU w/o mean	64.5	16.6	35.4	18.7
	+ MCU	64.5	22.8	26.4	**3.6**

Sensitivity to the Projection Rank K. We additionally sweep K while fixing all other hyperparameters. Each run uses MLP Breaking + CIR + MCU. Highlighted rows in Table 4 are the dataset-specific settings used in Table 1: $K = 32$ on WMDP-Cyber and $K = 8$ on WMDP-Bio. Utility remains stable throughout the sweep. WMDP-Cyber is relatively insensitive to K. WMDP-Bio is weaker at $K = 2$, but improves substantially thereafter and remains robust over the broader range from $K = 4$ to $K = 32$.

6 Conclusion

We investigated the fragility of LLM unlearning from a representation geometry perspective and identified a fundamental mechanism: existing unlearning methods predominantly modify dominant components of internal representations, which are easily recovered during relearning attacks. In contrast, minor components exhibit significantly stronger resistance to such recovery. Building on this insight, we proposed MCU, a method that explicitly targets the robust minor component subspace by projecting out dominant directions before computing unlearning losses. MCU is compatible with existing representation-based unlearning objectives and complementary to gradient-level filtering techniques like CIR. Extensive experiments on WMDP-Cyber, WMDP-Bio, and Years datasets demonstrate that MCU significantly reduces knowledge recovery under relearning attacks while preserving model utility.

AI USE STATEMENT

In this work, we used generative AI tools for writing, editing, and formatting. We did not use generative AI tools as part of the core method, experiments, or scientific claims. Other required-disclosure uses are not applicable to this work. We take responsibility for the final content of this work.

ETHICS STATEMENT

This work does not involve human subjects or crowdsourcing, and it does not release new model checkpoints or hazardous datasets. Appendix B discusses positive uses for persistent removal of safety-relevant knowledge, negative uses such as suppressing beneficial knowledge or safety-aligned behavior, the risk of over-trusting unlearning, and corresponding deployment mitigations. Appendix A states the scope of the evaluation and the assumptions of the theory.

REFERENCES

Clark Barrett, Brad Boyd, Elie Bursztein, Nicholas Carlini, Brad Chen, Jihye Choi, Amrita Roy Chowdhury, Mihai Christodorescu, Anupam Datta, Soheil Feizi, et al. Identifying and mitigating the security risks of generative ai. *Foundations and Trends in Privacy and Security*, 6(1):1–52, 2023.

Kushal Raj Bhandari, Pin-Yu Chen, and Jianxi Gao. Forecasting open-weight ai model growth on huggingface. *arXiv preprint arXiv:2502.15987*, 2025.

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

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480. IEEE, 2015.

Stephen Casper, Kyle O'Brien, Shayne Longpre, Elizabeth Seger, Kevin Klyman, Rishi Bommasani, Aniruddha Nrusimha, Ilia Shumailov, Sören Mindermann, Steven Basart, et al. Open technical problems in open-weight ai model risk management. *Social Science Research Network*, 2025.

Aghyad Deeb and Fabien Roger. Do unlearning methods remove information from language model weights? *arXiv preprint arXiv:2410.08827*, 2024.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms, 2023.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gn0mIhQGNM.

Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. Towards llm unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. In *Forty-second International Conference on Machine Learning*, 2025.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9304–9312, 2020.

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

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2): 217–288, 2011.

Shengyuan Hu, Yiwei Fu, Steven Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned LLMs via benign relearning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fMNRYBvcQN.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, 2023.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36:1957–1987, 2023.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pp. 1–14, 2025.

Jakub Łucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. An adversarial perspective on machine unlearning for ai safety. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=J5IRyTKZ9s.

Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *arXiv preprint arXiv:2402.16835*, 2024.

M-A-P, Ge Zhang, Xinrun Du, Zhimiao Yu, Zili Wang, Zekun Wang, Shuyue Guo, Tianyu Zheng, Kang Zhu, Jerry Liu, Shawn Yue, Binbin Liu, Zhongyuan Peng, Yifan Yao, Jack Yang, Ziming Li, Bingni Zhang, Minghao Liu, Tianyu Liu, Yang Gao, Wenhu Chen, Xiaohuan Zhou, Qian Liu, Taifeng Wang, and Wenhao Huang. Finefineweb: A comprehensive study on fine-grained domain web corpus, December 2024. URL https://huggingface.co/datasets/m-a-p/FineFineWeb.

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

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2408–2417. PMLR, 2015. URL https://proceedings.mlr.press/v37/martens15.html.

Nestor Maslej, Loredana Fattorini, Raymond Perrault, Vanessa Parli, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, and Jack Clark. Artificial intelligence index report 2024. Technical report, Stanford Institute for Human-Centered Artificial Intelligence (HAI), 2024. URL https://hai.stanford.edu/ai-index/2024-ai-index-report. Seventh edition. Available as AI Index Report via arXiv:2405.19522.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Neel Nanda, Senthooran Rajamanoharan, Janos Kramar, and Rohin Shah. Fact finding: Attempting to reverse-engineer factual recall on the neuron level, Dec 2023. URL https://www.alignmentforum.org/posts/iGuwZTHWb6DFY3sKB.

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–46, 2025.

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

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to!, 2023. URL https://arxiv.org/abs/2310.03693.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. Representation noising: A defence mechanism against harmful finetuning. *Advances in Neural Information Processing Systems*, 37:12636–12676, 2024.

Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, et al. Latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*, 2024.

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

Filip Sondej and Yushi Yang. Collapse of irrelevant representations (cir) ensures robust and non-disruptive llm unlearning. *arXiv preprint arXiv:2509.11816*, 2025.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, et al. Tamper-resistant safeguards for open-weight llms. *arXiv preprint arXiv:2408.00761*, 2024.

Pratiksha Thaker, Yash Maurya, Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. Guardrail baselines for unlearning in llms. *arXiv preprint arXiv:2403.03329*, 2024.

Enayat Ullah, Tung Mai, Anup Rao, Ryan A Rossi, and Raman Arora. Machine unlearning via algorithmic stability. In *Conference on Learning Theory*, pp. 4126–4142. PMLR, 2021.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024.

A Limitations

Our work has several aspects worth noting. First, our experimental evaluation focuses on relearning attacks, which represent the most practically relevant threat for open-weight models. Robustness against complementary attack vectors—such as inference-time jailbreaking (Łucki et al., 2025; Lynch et al., 2024) or quantization-induced knowledge revival (Qi et al., 2023)—may benefit from additional

defenses, and evaluating MCU against these settings is a natural direction for future work. Second, our theoretical analysis in Section 3.2 employs a first-order (NTK-regime) linearization for tractability; tightening this framework to capture non-linear dynamics more precisely, or establishing formal convergence guarantees for the MCU objective, are interesting open theoretical questions.

B Broader Impacts

Our research advances the robustness of LLM unlearning against relearning attacks, which is critical for ensuring that safety-relevant knowledge removal is persistent in open-weight models. By revealing the representation-level mechanism underlying unlearning fragility and proposing a principled solution, this work contributes to more reliable privacy protection and regulatory compliance for deployed language models.

At the same time, stronger and more persistent unlearning can have negative uses if applied to suppress beneficial knowledge, remove safety-aligned behaviors, or conceal model provenance and accountability-relevant information. There is also a risk that practitioners over-trust unlearning as a complete safety guarantee, even though our evaluation focuses on relearning attacks and does not cover every possible recovery channel. Responsible deployment should therefore combine robust unlearning with independent audits, explicit retain-set and safety evaluations, access controls for high-risk settings, and monitoring for both accidental utility degradation and intentional misuse. We do not release new model checkpoints, hazardous datasets, or other high-risk assets as part of this submission.

C Related Works

LLM Unlearning. Machine unlearning, originally developed to address post-training privacy concerns such as the "right to be forgotten" (Cao & Yang, 2015; Ginart et al., 2019; Ullah et al., 2021), aims to modify trained models to remove the influence of specific data without costly retraining. While approximate unlearning methods have been successfully applied in various domains (Kurmanji et al., 2023; Bourtoule et al., 2021; Nguyen et al., 2025; Golatkar et al., 2020; Jia et al., 2023; Fan et al., 2024), LLM unlearning has emerged as a rapidly growing subfield (Jang et al., 2023; Yao et al., 2024; Eldan & Russinovich, 2023; Zhang et al., 2024; Maini et al., 2024; Liu et al., 2025) that aims to remove undesired data influences from large language models while preserving model utility for unrelated tasks. Applications span mitigating harmful content generation (Yao et al., 2024; Li et al., 2024), protecting copyrighted and private information (Eldan & Russinovich, 2023; Jang et al., 2023), and preventing LLMs from producing biosecurity or cybersecurity threats (Li et al., 2024; Barrett et al., 2023). Current approaches fall into two categories: **model optimization-based methods** (Maini et al., 2024; Yao et al., 2024; Zhang et al., 2024; Li et al., 2024) that fine-tune model parameters, and **input-based strategies** that leverage prompting or in-context learning to suppress undesired behaviors (Thaker et al., 2024; Pawelczyk et al., 2024). Several benchmarks have been proposed to evaluate unlearning effectiveness, including TOFU (Maini et al., 2024) for fictitious unlearning, WMDP (Li et al., 2024) for hazardous knowledge removal, and MUSE (Shi et al., 2024) for copyright protection. Among existing methods, NPO (Zhang et al., 2024) has emerged as a promising approach by framing unlearning as preference optimization.

Robustness Challenges in LLM Unlearning. Recent studies have exposed critical vulnerabilities in existing LLM unlearning methods (Lynch et al., 2024; Łucki et al., 2025; Hu et al., 2025; Deeb & Roger, 2024). These vulnerabilities primarily manifest through two attack categories: **relearning attacks** (Hu et al., 2025; Lynch et al., 2024; Deeb & Roger, 2024), where fine-tuning with even a small subset of forget samples can restore unlearned knowledge; and **jailbreaking attacks** (Łucki et al., 2025; Lynch et al., 2024), where adversarial prompts successfully recover forgotten information at inference time. Even unrelated operations such as model quantization can inadvertently revive targeted knowledge (Qi et al., 2023). Most alarmingly, Deeb & Roger (2024) demonstrated that current unlearning methods achieve recovery rates exceeding 88% after relearning attacks, indicating that knowledge is merely hidden rather than truly removed from model weights. To address these robustness challenges, recent work has explored various defense strategies. Tamirisa et al. (2024) leveraged model-agnostic meta-learning (MAML) to counter tampering attacks, while Sheshadri et al. (2024) employed adversarial training in the latent space of LLMs. Sondej & Yang (2025) proposed

CIR, which uses PCA to identify and remove common representations from unlearning gradients before applying updates. From an optimization perspective, Fan et al. (2025) investigated SAM to improve unlearning robustness through smoother loss landscapes. Despite these advances, the fundamental mechanism underlying unlearning fragility remains poorly understood, motivating our representation-centric analysis.

D Theoretical Analysis: Full Derivations

This appendix provides the detailed derivations supporting Theorems 1 and 2 in Section 3.2. Throughout, we focus on a single MLP module with input-side activation $\mathbf{h}_\theta(\mathbf{x}) \in \mathbb{R}^d$; the argument applies layerwise.

D.1 Notation and Linearization

Let $\mathbf{x}$ be a token (or sequence) drawn from the forget distribution $\mathcal{D}_f$, and let $\mathbf{h}_o(\mathbf{x})$ be the original (pre-unlearning) representation. After centering, the representation covariance is

$$\mathbf{\Sigma} \;=\; \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_f}\big[\mathbf{h}_o(\mathbf{x})\mathbf{h}_o(\mathbf{x})^\top\big] \;=\; \sum_{k=1}^{d} \sigma_k^2\, \mathbf{v}_k\mathbf{v}_k^\top, \qquad \sigma_1^2 \geq \cdots \geq \sigma_d^2. \tag{16}$$

Observation 1 corresponds to a sharp decay of σ_k^2 in k. In a small neighborhood of the pre-unlearning parameters θ_o, we use the first-order expansion

$$\mathbf{h}_\theta(\mathbf{x}) \;\approx\; \mathbf{h}_o(\mathbf{x}) + \mathbf{J}(\mathbf{x})\,\Delta\theta, \qquad \mathbf{J}(\mathbf{x}) = \left.\frac{\partial \mathbf{h}_\theta(\mathbf{x})}{\partial \theta}\right|_{\theta_o}. \tag{17}$$

Define the empirical neural tangent kernel (Jacot et al., 2018)

$$\mathbf{K}(\mathbf{x},\mathbf{x}') \;=\; \mathbf{J}(\mathbf{x})\,\mathbf{J}(\mathbf{x}')^\top \in \mathbb{R}^{d\times d}. \tag{18}$$

In the lazy/NTK regime, after standard normalization, $\mathbf{K}(\mathbf{x},\mathbf{x}') \approx \kappa\,\mathbf{I}$ for some $\kappa > 0$, approximately independent of the inputs. We use this approximation only to expose the dominant scaling; the conclusions are stable to mild anisotropy in $\mathbf{K}$.

D.2 A Unified Form for Unlearning Losses

We treat both representation-level losses (RMU, MLP Breaking) and output-level losses (GA, NPO) under a single framework. Let $\mathbf{h}_\theta(\mathbf{x}) \in \mathbb{R}^d$ be the analyzed intermediate representation (e.g., an MLP activation in the layer where PCA is performed). Any unlearning objective can be written as

$$\mathcal{L}_u(\theta) \;=\; \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_f}\big[\,\ell_u(f_\theta(\mathbf{x});\,\mathbf{x})\,\big], \tag{19}$$

where $f_\theta(\mathbf{x})$ denotes any output of the model that depends on θ through $\mathbf{h}_\theta(\mathbf{x})$. By the chain rule,

$$\nabla_\theta \mathcal{L}_u \;=\; \mathbb{E}_{\mathbf{x}}\big[\mathbf{J}(\mathbf{x})^\top \mathbf{g}_u(\mathbf{x})\big], \qquad \mathbf{g}_u(\mathbf{x}) \;:=\; \left.\frac{\partial \ell_u}{\partial \mathbf{h}}\right|_{\mathbf{h}_o(\mathbf{x})}, \tag{20}$$

i.e., *any* unlearning loss is mediated by an effective per-sample residual $\mathbf{g}_u(\mathbf{x})$ in the representation space of $\mathbf{h}$.

Lemma 1 (Eigenstructure of effective residuals)**.** *For all four unlearning losses studied in this paper (RMU, MLP Breaking, GA, NPO), the effective residual covariance admits the decomposition*

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_f}\big[\mathbf{g}_u(\mathbf{x})\mathbf{g}_u(\mathbf{x})^\top\big] \;=\; \mathbf{A}\,\mathbf{\Sigma}\,\mathbf{A}^\top \;+\; \mathbf{N}, \tag{21}$$

where $\mathbf{A}$ is a loss-dependent linear map and $\mathbf{N} \succeq 0$ has small operator norm $\|\mathbf{N}\| = O(\tau^2) \ll \sigma_1^2$. Moreover, on the dominant subspace spanned by $\{\mathbf{v}_1,\ldots,\mathbf{v}_K\}$ (the few PCs that carry essentially all variance) the map $\mathbf{A}$ approximately preserves the eigenbasis of $\mathbf{\Sigma}$, in the sense that

$$\mathbf{v}_k^\top \mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top \mathbf{v}_k \;=\; \sum_{\ell} \sigma_\ell^2 \left(\mathbf{v}_k^\top \mathbf{A}\,\mathbf{v}_\ell\right)^2 \;=\; \alpha_k\,\sigma_k^2 \;+\; O(\tau^2), \quad \alpha_k > 0, \tag{22}$$

for $k \leq K$. Consequently the eigenvalues of $\mathbb{E}[\mathbf{g}_u\mathbf{g}_u^\top]$ aligned with the top-k subspace of $\mathbf{\Sigma}$ are $\Theta(\sigma_k^2)$.

Proof. We verify the lemma case by case. The decomposition $\mathbb{E}[\mathbf{g}_u\mathbf{g}_u^\top] = \mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top + \mathbf{N}$ is shown for each loss; the diagonal-on-dominant-subspace property (22) is then justified.

Representation-level losses (RMU, MLP Breaking). These directly act on $\mathbf{h}$, so $f_\theta(\mathbf{x}) = \mathbf{h}_\theta(\mathbf{x})$. With quadratic surrogate $\ell_u(\mathbf{h};\mathbf{x}) = \frac{1}{2}\|\mathbf{h} - \mathbf{t}(\mathbf{x})\|^2$, $\mathbf{g}_u(\mathbf{x}) = \mathbf{h}_o(\mathbf{x}) - \mathbf{t}(\mathbf{x})$. Thus $\mathbf{A} = \mathbf{I}$, and

$$\mathbb{E}[\mathbf{g}_u\mathbf{g}_u^\top] = \mathbf{\Sigma} - \mathbb{E}[\mathbf{h}_o\mathbf{t}^\top] - \mathbb{E}[\mathbf{t}\mathbf{h}_o^\top] + \mathbb{E}[\mathbf{t}\mathbf{t}^\top] = \mathbf{\Sigma} + \mathbf{N}.$$

For RMU's random-direction targets and MLP Breaking's noise targets, $\mathbf{t}(\mathbf{x})$ is independent of $\mathbf{h}_o(\mathbf{x})$, so the cross terms vanish in expectation and $\|\mathbf{N}\| = \|\mathbb{E}[\mathbf{t}\mathbf{t}^\top]\| = O(\tau^2)$. Since $\mathbf{A} = \mathbf{I}$ commutes with $\mathbf{\Sigma}$, (22) holds exactly with $\alpha_k = 1$.

Output-level loss: Gradient Ascent (GA). Here $f_\theta(\mathbf{x}) = \mathbf{z}_\theta(\mathbf{x})$ are the logits, with $\mathbf{z} = \mathbf{W}_{\text{out}}\,\boldsymbol{\phi}(\mathbf{h}) + \mathbf{b}$ for some downstream nonlinearity $\boldsymbol{\phi}$ and head $\mathbf{W}_{\text{out}}$. The GA loss is $\ell_u = +\log p_\theta(y \mid \mathbf{x})$, whose gradient w.r.t. $\mathbf{h}$ is

$$\mathbf{g}_u(\mathbf{x}) \;=\; \mathbf{D}(\mathbf{x})\,\mathbf{W}_{\text{out}}^\top\big(\mathbf{e}_{y(\mathbf{x})} - \mathbf{p}(\mathbf{x})\big),$$

where $\mathbf{p}(\mathbf{x}) = \mathrm{softmax}(\mathbf{z}_o(\mathbf{x}))$ and $\mathbf{D}(\mathbf{x}) = \partial\boldsymbol{\phi}/\partial\mathbf{h}|_{\mathbf{h}_o}$. We linearize $\mathbf{g}_u$ around the population mean $\bar{\mathbf{h}} = \mathbb{E}[\mathbf{h}_o]$. Since both $\mathbf{D}(\mathbf{x})$ and the residual $\mathbf{e}_{y(\mathbf{x})} - \mathbf{p}(\mathbf{x})$ depend on $\mathbf{x}$ only through $\mathbf{h}_o(\mathbf{x})$ (the labels $y(\mathbf{x})$ being deterministic given the prefix at a well-trained checkpoint), a first-order Taylor expansion gives

$$\mathbf{g}_u(\mathbf{x}) \;=\; \bar{\mathbf{g}} \;+\; \mathbf{A}\,(\mathbf{h}_o(\mathbf{x}) - \bar{\mathbf{h}}) \;+\; \mathbf{r}(\mathbf{x}), \qquad \mathbf{A} \;:=\; \left.\frac{\partial \mathbf{g}_u}{\partial \mathbf{h}_o}\right|_{\bar{\mathbf{h}}}, \tag{23}$$

with remainder $\|\mathbf{r}(\mathbf{x})\| = O(\|\mathbf{h}_o - \bar{\mathbf{h}}\|^2)$. Centering ($\bar{\mathbf{g}}$ is absorbed into a mean-correction term that yields $O(\tau^2)$ contribution after centering $\mathbf{h}_o$) and using the centering of $\mathbf{h}_o$ assumed in Appendix D.1 yields

$$\mathbb{E}[\mathbf{g}_u\mathbf{g}_u^\top] \;=\; \mathbf{A}\,\mathbf{\Sigma}\,\mathbf{A}^\top \;+\; \mathbf{N}, \qquad \|\mathbf{N}\| = O(\tau^2),$$

where τ^2 collects (i) the squared remainder and (ii) the residual magnitude $\|\mathbf{e}_y - \mathbf{p}\|^2$, which is small because the model has already fit $\mathcal{D}_f$ at the pre-unlearning checkpoint.

Output-level loss: NPO. NPO's loss (2) is a sigmoid-shaped reweighting of the cross-entropy on $\mathcal{D}_f$ vs. a reference model:

$$\ell_u^{\text{NPO}}(\mathbf{x}) \;=\; -\frac{2}{\beta}\,\log\sigma\big(-\beta\,\big[\log p_\theta(y \mid \mathbf{x}) - \log p_{\text{ref}}(y \mid \mathbf{x})\big]\big)\,.$$

Its representation gradient is $\mathbf{g}_u^{\text{NPO}}(\mathbf{x}) = w_\beta(\mathbf{x})\,\mathbf{g}_u^{\text{GA}}(\mathbf{x})$ with $w_\beta(\mathbf{x}) = 2\,\sigma(\beta[\log p_\theta - \log p_{\text{ref}}]) \in (0,2)$. Linearizing w_β around its mean as in (23) gives $\mathbf{g}_u^{\text{NPO}} = \bar{w}\,\mathbf{g}_u^{\text{GA}} + O(\|\mathbf{h}_o - \bar{\mathbf{h}}\|^2)$, so

$$\mathbb{E}[\mathbf{g}_u^{\text{NPO}}(\mathbf{g}_u^{\text{NPO}})^\top] \;=\; \bar{w}^2\,\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top \;+\; \mathbf{N}', \qquad \|\mathbf{N}'\| = O(\tau^2),$$

i.e. the same map $\mathbf{A}$ scaled by a positive constant.

Diagonal property (22) **on the dominant subspace.** For RMU/MLP Breaking, $\mathbf{A} = \mathbf{I}$ and (22) is exact. For GA/NPO, observe that the principal components $\{\mathbf{v}_k\}_{k\le K}$ are estimated from activations *at the same layer* where $\mathbf{g}_u$ lives; the well-trained head $\mathbf{W}_{\text{out}}$ together with the diagonal-by-construction nonlinearity $\boldsymbol{\phi}$ and the softmax linearization tend to align $\mathbf{A}$'s singular vectors with the dominant subspace of $\mathbf{\Sigma}$ (otherwise the model could not have achieved low loss using only those directions). Concretely, decomposing $\mathbf{A} = \mathbf{V}\,\mathrm{diag}(\alpha)\,\mathbf{V}^\top + \mathbf{E}$ with $\mathbf{V} = [\mathbf{v}_1,\dots,\mathbf{v}_d]$ and small off-diagonal $\mathbf{E}$, we get $\mathbf{v}_k^\top\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top\mathbf{v}_k = \alpha_k^2\sigma_k^2 + \sum_{\ell\neq k}\sigma_\ell^2(\mathbf{v}_k^\top\mathbf{E}\mathbf{v}_\ell)^2$, and the second term is dominated by $\|\mathbf{E}\|^2\sigma_1^2 = O(\tau^2)$ provided $\mathbf{A}$ is approximately diagonal in the PCA basis. Even without this assumption, a Weyl-type inequality gives $\lambda_k(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top) \ge \sigma^2_{\min,K}(\mathbf{A})\,\sigma_k^2$ where $\sigma_{\min,K}(\mathbf{A})$ is the smallest singular value of $\mathbf{A}$ restricted to the top-K subspace, so the qualitative scaling $\Theta(\sigma_k^2)$ on the dominant subspace is unchanged. □

Lemma 1 is the technical bridge that lets the same NTK-spectrum argument apply uniformly across representation-level and output-level losses. We restate the proofs of Theorems 1 and 2 in this generality below.

D.3 Proof of Theorem 1 (Dominant-Component Concentration)

Unlearning is run with mini-batch SGD: at each step a sample $\mathbf{x}_t \sim \mathcal{D}_f$ produces the stochastic update $\Delta\theta_t = -\eta\, \mathbf{J}(\mathbf{x}_t)^\top \mathbf{g}_u(\mathbf{x}_t)$. Substituting into the linearized representation and evaluating at any forget-set point $\mathbf{x}'$:

$$\Delta\mathbf{h}_t(\mathbf{x}') \;=\; \mathbf{J}(\mathbf{x}')\,\Delta\theta_t \;=\; -\eta\, \mathbf{K}(\mathbf{x}', \mathbf{x}_t)\, \mathbf{g}_u(\mathbf{x}_t). \tag{24}$$

Under $\mathbf{K}(\mathbf{x}', \mathbf{x}_t) \approx \kappa \mathbf{I}$ this becomes $\Delta\mathbf{h}_t(\mathbf{x}') \approx -\eta\kappa\, \mathbf{g}_u(\mathbf{x}_t)$. Note that we keep the *stochastic, per-sample* residual rather than the population mean: taking expectation *before* squaring (as in $\mathbb{E}[\mathbf{g}_u]$) would mix signal and noise scales incorrectly. Instead, the appropriate quantity for the change-ratio metric (5), which is computed from squared inner products averaged over $\mathbf{x}'$, is the *expected squared per-sample displacement*.

Projecting onto $\mathbf{v}_k$ and squaring:

$$\mathbb{E}_{\mathbf{x}_t}\big[\langle \Delta\mathbf{h}_t(\mathbf{x}'), \mathbf{v}_k\rangle^2\big] \;\approx\; \eta^2\kappa^2\, \mathbb{E}_{\mathbf{x}_t}\big[\langle \mathbf{g}_u(\mathbf{x}_t), \mathbf{v}_k\rangle^2\big] \;=\; \eta^2\kappa^2\, \mathbf{v}_k^\top \mathbb{E}[\mathbf{g}_u \mathbf{g}_u^\top]\, \mathbf{v}_k. \tag{25}$$

By Lemma 1 (in particular (22)),

$$\mathbf{v}_k^\top \mathbb{E}[\mathbf{g}_u \mathbf{g}_u^\top]\, \mathbf{v}_k \;=\; \alpha_k\, \sigma_k^2 + O(\tau^2), \tag{26}$$

for a loss-dependent positive constant α_k that is bounded away from 0 on the dominant subspace. Accumulating T small i.i.d. steps, the squared displacement averaged over the forget set scales as

$$\mathbb{E}_{\mathcal{D}_f}\big[\langle \mathbf{h}_\mathrm{u} - \mathbf{h}_\mathrm{o}, \mathbf{v}_k\rangle^2\big] \;\propto\; T\,\sigma_k^2 \;+\; O(\tau^2), \tag{27}$$

matching Equation 7. Equivalently, the typical absolute displacement scales as $\sqrt{T}\,\sigma_k$, so the un-normalized numerator of the change-ratio (5) mirrors the singular-value (square-root explained-variance) profile of $\mathbf{\Sigma}$. The argument is identical for representation-level and output-level losses; the only loss-specific quantity is the constant prefactor α_k, which does not affect the qualitative scaling. □

D.4 Proof of Theorem 2 (Dominant-Component Recoverability)

Let $\mathcal{D}_r$ denote the relearning distribution used by the attacker. Under the standard threat model, $\mathcal{D}_r$ is structurally similar to $\mathcal{D}_f$, so

$$\mathbf{\Sigma}_r \;=\; \mathbb{E}_{\mathcal{D}_r}[\mathbf{h}_o \mathbf{h}_o^\top] \;\approx\; \sum_k \tilde{\sigma}_k^2\, \mathbf{v}_k \mathbf{v}_k^\top, \qquad \tilde{\sigma}_k^2 \asymp \sigma_k^2. \tag{28}$$

Standard relearning attacks use a maximum-likelihood (cross-entropy) objective on $\mathcal{D}_r$. By the same chain-rule decomposition (20), the relearning gradient is $\nabla_\theta \mathcal{L}_r = \mathbb{E}_{\mathcal{D}_r}[\mathbf{J}(\mathbf{x})^\top \mathbf{g}_r(\mathbf{x})]$ with effective residual $\mathbf{g}_r(\mathbf{x})$. The same case analysis as in Lemma 1 (specialized to GA-style cross-entropy on $\mathcal{D}_r$) gives $\mathbb{E}[\mathbf{g}_r \mathbf{g}_r^\top] = \mathbf{A}_r \mathbf{\Sigma}_r \mathbf{A}_r^\top + O(\tau^2)$, with $\mathbf{A}_r$ approximately diagonal in the PCA basis on the dominant subspace. Iterating the linearized dynamics over T_r relearning steps with constant step size, the projection of $\mathbf{h}_u - \mathbf{h}_r$ onto $\mathbf{v}_k$ follows an exponential approach to the original $\mathbf{h}_o$ projection with rate $c\sigma_k^2$:

$$\text{Recovery Ratio}_k \;=\; \frac{\langle \mathbf{h}_u - \mathbf{h}_r, \mathbf{v}_k\rangle}{\langle \mathbf{h}_u - \mathbf{h}_o, \mathbf{v}_k\rangle} \;\approx\; 1 - \exp\big(-c\,\sigma_k^2\, T_r\big). \tag{29}$$

The result depends on neither the specific unlearning loss used to produce $\mathbf{h}_u$ (since the recovery ratio is normalized by the actual unlearning displacement) nor the specific relearning loss family (any loss whose effective residual covariance shares the dominant eigenstructure of $\mathbf{\Sigma}_r$ yields the same scaling). Reaching a fixed recovery level $1-\delta$ on direction $\mathbf{v}_k$ thus requires $T_r = O\big(\sigma_k^{-2} \log(1/\delta)\big)$ steps; dominant components saturate to 1 within a few attack steps, while minor components require an *inverse-variance* blow-up in the number of steps. □

D.5 Why Minor Components Are Structurally Hard to Recover

The inverse-variance recovery scaling above explains *rate* differences but not why minor components fail to recover even when the attacker invests a large T_r. We complete the picture with a signal-to-noise (SNR) argument on the relearning gradient.

Decompose the per-sample representation as $\mathbf{h}_o(\mathbf{x}) = \sum_k a_k(\mathbf{x})\,\mathbf{v}_k$ with coefficients $a_k(\mathbf{x}) = \langle \mathbf{h}_o(\mathbf{x}), \mathbf{v}_k \rangle$, so $\mathbb{E}[a_k(\mathbf{x})] = 0$ and $\mathbb{E}[a_k(\mathbf{x})^2] = \sigma_k^2$ by construction. Because the coefficients are mean-zero, a naive cross-sample correlation $\mathbb{E}_{\mathbf{x},\mathbf{x}'}[a_k(\mathbf{x})a_k(\mathbf{x}')]$ across *independent* samples vanishes identically and conveys no information; instead, agreement must be measured *conditionally* on shared structure. Concretely, let c denote a latent context variable (e.g., the topic, document, or local sub-distribution from which $\mathbf{x}$ is drawn) and decompose

$$a_k(\mathbf{x}) \;=\; s_k(c) \;+\; \epsilon_k(\mathbf{x}), \qquad s_k(c) := \mathbb{E}[a_k(\mathbf{x}) \mid c], \quad \mathbb{E}[\epsilon_k \mid c] = 0. \tag{30}$$

We define the (shared-signal) cross-sample agreement of the k-th coordinate as the fraction of variance carried by the context-shared component:

$$\rho_k \;=\; \frac{\mathrm{Var}_c\big(s_k(c)\big)}{\sigma_k^2} \;=\; \frac{\mathbb{E}_{\mathbf{x},\mathbf{x}'|c}\big[a_k(\mathbf{x})\,a_k(\mathbf{x}')\big]}{\sigma_k^2}, \tag{31}$$

where the second equality holds when $\mathbf{x}, \mathbf{x}'$ are drawn *conditional on the same context* c. Equivalently, ρ_k is the intra-class/inter-class variance ratio along $\mathbf{v}_k$, and $\sqrt{\rho_k}$ is the cosine alignment between the per-sample gradient $\nabla_\theta \ell_r(\mathbf{x})$ and its batch average projected onto $\mathbf{v}_k$. With this definition, a finite-batch relearning gradient along $\mathbf{v}_k$ has signal $\propto \sigma_k\sqrt{\rho_k}$ and noise $\propto \sigma_k\sqrt{(1-\rho_k)/B}$ for batch size B, giving an SNR of order $\sqrt{B\rho_k/(1-\rho_k)}$. Two regimes emerge:

- **Dominant components**: $\rho_k \to 1$, since these directions encode features shared between $\mathcal{D}_r$ and $\mathcal{D}_f$ at the topic/context level (e.g., topical regularities, syntactic patterns). Relearning gradients along $\mathbf{v}_k$ accumulate coherently across the batch, and recovery proceeds at the rate predicted by Theorem 2.
- **Minor components**: $\rho_k \approx 0$, since these directions encode sample-specific structure that varies idiosyncratically within each context. The batched relearning gradient averages out, the SNR collapses, and no amount of attacker fine-tuning on *related* data can reliably reconstruct the minor-component values that the original model used for the held-out forget samples.

This formalizes the intuition stated after Observation 3 and is consistent with both Figure 2c and the cross-loss replications in Appendix L.

D.6 Discussion of Assumptions

The two assumptions used above merit comment. **(i) NTK linearization.** The lazy-regime approximation $\mathbf{K}(\mathbf{x}, \mathbf{x}') \approx \kappa\mathbf{I}$ is exact only in the infinite-width limit; in practice it is a useful first-order approximation when the unlearning step size and number of steps remain modest, which is precisely the regime in which fragile unlearning operates (otherwise utility on retain data collapses). The qualitative conclusions—change-ratio scaling with σ_k^2 and exponential recovery with rate σ_k^2—survive any anisotropy in $\mathbf{K}$ that is not specifically aligned against the dominant subspace. **(ii) Effective-residual eigenstructure (Lemma 1).** The proof verified this for RMU, MLP Breaking, GA, and NPO. For representation-level losses, the assumption reduces to the target $\mathbf{t}(\mathbf{x})$ being uncorrelated with the dominant subspace, which holds for the random/noise/zero targets used in practice. For output-level losses, the assumption follows because (a) the model's predictions on $\mathcal{D}_f$ depend on $\mathbf{x}$ only through $\mathbf{h}_o(\mathbf{x})$, so any sample-to-sample variation in the output residual is mediated by $\boldsymbol{\Sigma}$, and (b) the post-$\mathbf{h}$ Jacobian $\mathbf{D}\mathbf{W}_{\text{out}}^\top$ is full-rank on the dominant subspace at any well-trained checkpoint. The empirical universality of Observation 2 across unlearning losses (Appendix L) confirms that the assumption holds in practice for both representation-level and output-level losses.

E Spectral Recovery, MCU, and the Connection to CIR

Spectral recovery as a mechanism of unlearning fragility. CIR (Sondej & Yang, 2025) motivates principal-component gradient filtering through the intuition that unlearning disrupts shared representations that subsequent fine-tuning can restore. However, it does not provide a empirical or theoretical analysis of direction-wise representation changes and their recovery dynamics. Our work studies these dynamics directly: unlearning concentrates changes in dominant representation components, and relearning preferentially recovers those changes (Section 3). As shown in Theorems 1 and 2, our analysis connects this asymmetry to the representation spectrum. This identifies the allocation of unlearning changes across directions with different recovery rates as a mechanism governing robustness.

MCU as a theory-guided intervention. This mechanism motivates MCU: a representation-level loss that suppresses dominant components, directing optimization pressure toward minor components. MCU makes the spectral allocation of forgetting part of the objective itself. Our representation analysis and subspace controls (Figure 3 and Table 2) support this design by showing a redistribution of changes toward minor components and lower post-attack accuracy with the proposed projection. The contribution therefore comprises both an account of spectral recovery and a method designed from that account.

A curvature-based interpretation of CIR. Although Sondej & Yang (2025) motivates CIR with the intuition that unlearning disrupts shared representations, the method does not follow explicitly or necessarily from this intuition. To explain this discrepancy, we provide a curvature-based interpretation that explains how CIR works and why it is effective. CIR (Sondej & Yang, 2025) can be interpreted as redirecting updates toward directions with lower relearning curvature, where unlearning changes decay more slowly during relearning. To make this connection explicit, consider a linear module $z = Wa$ with input activation a and output gradient $g_u = \partial \ell_u / \partial z$. CIR filters both vectors before forming the weight update. Modeling these filters as orthogonal projections P_a and P_g onto the retained subspaces gives

$$\Delta W_{\mathrm{CIR}} = -\eta_u (P_g g_u)(P_a a)^\top = -\eta_u P_g (g_u a^\top) P_a. \tag{32}$$

To identify the curvature of the retained modes, we approximate the local relearning Hessian block by its Fisher/GGN counterpart and then apply K-FAC (Martens & Grosse, 2015):

$$H_{r,W} \approx F_{r,W} \approx \widetilde{H}_{r,W} := A_r \otimes G_r, \qquad A_r = \mathbb{E}_r[aa^\top]. \tag{33}$$

Here $G_r = \mathbb{E}_r[g_r g_r^\top]$ is the module-output Fisher factor, where g_r is the module-output gradient of the negative log-likelihood. If $A_r u_i = \alpha_i u_i$ and $G_r w_j = \beta_j w_j$, with eigenvalues ordered in descending order, the parameter mode $w_j u_i^\top$ has approximate curvature $\lambda_{ij} = \alpha_i \beta_j$.

Suppose the activation and gradient subspaces removed by CIR align with the leading k_a and k_g eigendirections of these relearning factors. For the RTT attack, in which unlearning uses $T \cup V$ and relearning uses T, suppose the activation and gradient subspaces removed by CIR align with the leading k_a and k_g eigendirections of the relearning factors A_r and G_r on T, respectively. Then Equation 32 retains only modes with $i > k_a$ and $j > k_g$, whose curvatures satisfy

$$\lambda_{ij} \le \alpha_{k_a+1} \beta_{k_g+1}, \qquad i > k_a,\ j > k_g. \tag{34}$$

This bounds the curvature of the retained update subspace.

Taking the pre-unlearning parameters as a stationary recovery solution, let $e_{ij}(t)$ be the remaining unlearning displacement along mode $w_j u_i^\top$. In the fixed, layerwise quadratic model, ordinary gradient descent with sufficiently small step size gives

$$e_{ij}(T_r) = (1 - \eta_r \lambda_{ij})^{T_r} e_{ij}(0) \approx e^{-\eta_r \lambda_{ij} T_r} e_{ij}(0). \tag{35}$$

Consequently, the displacement in CIR's retained low-curvature modes loses only a small fraction of its initial magnitude when $\eta_r \alpha_{k_a+1} \beta_{k_g+1} T_r \ll 1$. Over an interval with fixed aligned filters, the accumulated unlearning displacement remains in this subspace. CIR therefore changes where forgetting occurs, placing it in modes that relearning reverses slowly under a limited budget.

Additional benefits beyond CIR. CIR filters parameter updates, while MCU specifies the representation directions penalized by the loss. Table 1 shows that adding MCU lowers post-attack Relearn accuracy for both RMU and MLP Breaking across all three datasets, with and without CIR. These comparisons demonstrate standalone gains and additional benefits on top of CIR. To control for differences in initial forgetting, Table 6 further compares checkpoints at matched Forget accuracy and comparable WikiText loss: adding MCU to MLP Breaking + CIR reduces Relearn accuracy from 38.9% to 32.6% on WMDP-Cyber and from 29.6% to 25.4% on WMDP-Bio. The additional robustness benefits thus persist at comparable initial forgetting levels, complementing the mechanistic contribution of our spectral analysis.

Summary. MCU and CIR act at different levels: MCU redirects unlearning changes toward minor representation components, while CIR filters parameter updates and, under the alignment assumptions above, favors low-curvature directions.

F Representation-Analysis Setup and Details

This appendix expands the representation-analysis protocol summarized in Section 3.1.

Modules and layers. We instrument the MLP `down_proj` output of every transformer block of Llama-3.1-8B (32 layers). For each layer we record activations at each token position of every example in the forget set $\mathcal{D}_{\mathrm{f}}$, yielding a tensor $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N \times d}$ per layer ℓ, where N is the total number of tokens in $\mathcal{D}_{\mathrm{f}}$ and $d = 4096$ is the hidden dimension. Results of other MLP sub-modules, shown in Appendix L, exhibit the same qualitative pattern.

PCA computation. Per layer, we center $\mathbf{H}^{(\ell)}$ by subtracting the per-coordinate mean and compute the principal components $\{\mathbf{v}_1^{(\ell)}, \ldots, \mathbf{v}_d^{(\ell)}\}$ via randomized SVD (Halko et al., 2011), with explained variances $\sigma_k^{(\ell)\,2}$. The same eigenbasis is used to project the unlearned and relearned activations $\mathbf{h}_u, \mathbf{h}_r$ collected on the same token positions. All ratios reported in Figures 2b and 2c are computed per layer and then averaged across layers.

G Experimental Details

Dataset details. For **WMDP-Cyber** and **WMDP-Bio**, we use the high-quality subsets of Sondej & Yang (2025) containing 203 cyber and 144 biological multiple-choice questions, each augmented with three short declarative sentences per question that together form the forget set used for unlearning. The **Years** dataset (Deeb & Roger, 2024) consists of 20th-century events paired with their dates. As retain sets we use FineFineWeb (M-A-P et al., 2024) subsets matched to the forget domain: `biology` for WMDP-Bio, `computer_science_and_technology` for WMDP-Cyber, and `fineweb-edu` for Years.

Baselines. Each evaluated forget loss paired with an specific retain loss to preserve model utility. For **Gradient Ascent (GA)** and **NPO** (Zhang et al., 2024), we apply the standard cross-entropy loss on the retain set. For **RMU** (Li et al., 2024) and **MLP Breaking**, following Li et al. (2024), we use a loss that penalizes the norm difference between the current and original model's representations on retain set, which encourages minimal deviation from the original model's representations.

Accuracy Computation. Following Sondej & Yang (2025), we compute accuracy as the expected probability of selecting the correct answer. Specifically, for multiple-choice questions with k options, we compute the probability distribution over answer choices using softmax with temperature $\tau = 1$ on the logits corresponding to the answer tokens. The accuracy for a batch is then computed as:

$$\text{Accuracy} = \frac{1}{|B|} \sum_{i \in B} p_i^{(\text{correct})}, \tag{36}$$

where $p_i^{(\text{correct})}$ denotes the probability assigned to the correct answer for sample i, and B is the batch. This expected accuracy metric provides a more fine-grained measure than hard accuracy (which only counts exact matches) and is more sensitive to partial knowledge changes during unlearning and relearning.

Relearning Attack Protocol. We follow the Retraining-on-T (RTT) attack protocol proposed by Deeb & Roger (2024). After unlearning on the full forget set $T \cup V$, we fine-tune the unlearned model on the training partition T (80% of the forget set) and evaluate accuracy recovery on the held-out validation partition V (20% of the forget set). For WMDP-Cyber and WMDP-Bio, we perform relearning for 100 epochs, while for the Years dataset we use 30 epochs due to its smaller size. To obtain robust estimates of post-attack accuracy, we follow Sondej & Yang (2025) and smooth the relearning accuracy curve by averaging over windows of 10 epochs for WMDP datasets and 3 epochs for Years. The reported **Relearn** accuracy corresponds to the maximum smoothed accuracy across the relearning trajectory, as some attack runs may exceed the optimal number of epochs.

Hyperparameter Selection for MCU. The key hyperparameter in our MCU method is K, the projection rank before computing the unlearning loss (Equation (11)): the mean direction plus the top

$K-1$ centered principal components. We perform a grid search over $K \in \{1, 2, 4, 8, 16, 32, 64\}$ and select the value that achieves the best trade-off between forget quality (low relearn accuracy). The main results use $K = 32$ on WMDP-Cyber and Years and $K = 8$ on WMDP-Bio; Table 4 reports the sweep on WMDP-Cyber and WMDP-Bio.

Unlearning Termination Criterion. Following Sondej & Yang (2025), we use the WikiText loss (Merity et al., 2016) as a criterion to determine when to terminate unlearning, in order to control for disruption to general language modeling performance. Specifically, we monitor the WikiText loss relative to its initial value before unlearning. Since different unlearning methods affect the WikiText loss differently, we use method-specific termination thresholds. The WikiText loss threshold primarily controls the number of training steps; we select thresholds for each method such that the unlearn accuracy approaches random chance (approximately 25% for 4-way multiple choice) while maintaining reasonable MMLU performance.

H Additional Results

Table 5 presents all experimental results across three datasets: WMDP-Cyber, WMDP-Bio, and Years.

I Comparisons at Matched Forget Accuracy

In Table 1, methods reach different Forget accuracies. Because Δ = Relearn − Forget is hard to interpret across different starting accuracies, those relearning gaps leave the comparison of recovered target knowledge unresolved. Held-out Relearn accuracy measures that recovered knowledge.

We trace both unlearning trajectories up to the same WikiText-loss budget, save every epoch, and independently apply RTT attack to every checkpoint. Table 6 then compares checkpoints with nearly identical Forget accuracy.

At the matched operating points, the two methods have essentially identical Forget accuracy and comparable WikiText loss. Adding MCU yields consistently lower post-attack Relearn accuracy and a smaller relearning gap Δ on both datasets. This shows that the robustness gain is preserved after controlling for initial Forget accuracy.

J Robustness under Adaptive Representation-Based Attacks

Because MCU operates on internal representations, a natural concern is whether an adversary that directly targets representations, rather than the standard RTT loss, can defeat it. We assume the attacker has access to the unlearned model, the original pre-unlearning model, and the forget set T, and fine-tunes the unlearned model to minimize the MSE between its MLP activations and those of the original model on T. This is a strictly stronger threat model than RTT, as it directly targets the representation-level modifications MCU introduces. Unlearning follows our main setup, and we report post-attack accuracy on the held-out split V.

Table 7 shows a clear hierarchy of robustness. GA collapses entirely under this attack, with relearn accuracy nearly returning to the original model, confirming that output-level unlearning leaves internal representations essentially intact and trivially recoverable. MLP Breaking + CIR is markedly more robust but still loses a non-trivial amount of forgotten knowledge, indicating that recoverable traces persist in the dominant components even after gradient-level filtering. Adding MCU yields by far the strongest robustness, with Δ several times smaller than MLP Breaking + CIR on both datasets. This trend matches our representation-geometry analysis (Section 3): the adaptive attacker can re-establish the cross-sample dominant-component structure, but cannot reliably reconstruct modifications in the minor-component subspace, which lacks the regularity needed for reconstruction.

Table 5: Complete experimental results on WMDP-Cyber, WMDP-Bio, and Years datasets. Highlighted rows are MCU variants; the grey rows are original-model baselines.

Dataset	Method	MMLU (↑)	WikiText (↓)	Forget (↓)	Relearn (↓)	Δ (↓)
WMDP-Cyber	Original model	65.1	1.000	57.6	-	-
	GA	61.3	1.503	25.1	57.0	31.9
	GA + SAM	60.1	1.100	27.4	57.7	30.3
	NPO	60.6	1.578	23.9	57.1	33.2
	NPO + SAM	60.1	1.101	27.7	57.7	30.0
	RMU	52.8	1.207	28.7	54.6	26.0
	RMU + SAM	52.7	1.201	28.3	53.5	25.1
	RMU + MCU	49.3	1.202	28.8	53.9	25.1
	RMU + CIR	64.1	1.006	28.0	50.5	22.5
	RMU + CIR + SAM	64.4	1.006	24.7	49.2	24.4
	RMU + CIR + MCU	64.7	1.006	31.7	43.3	11.6
	MLP Breaking	61.8	1.132	27.3	55.8	28.4
	MLP Breaking + SAM	58.0	1.215	25.6	47.4	21.8
	MLP Breaking + MCU	60.4	1.105	26.0	52.0	26.0
	MLP Breaking + CIR	65.0	1.001	20.2	37.2	17.0
	MLP Breaking + CIR + SAM	65.3	1.005	26.2	28.5	2.3
	MLP Breaking + CIR + MCU	64.7	1.001	25.9	31.6	5.7
WMDP-Bio	Original model	65.1	1.000	64.0	-	-
	GA	56.1	1.528	29.0	69.0	40.0
	GA + SAM	53.5	1.305	29.3	62.4	33.1
	NPO	51.9	1.419	25.9	66.7	40.8
	NPO + SAM	53.4	1.307	29.2	61.9	32.7
	RMU	45.5	1.300	26.7	49.8	23.1
	RMU + SAM	51.9	1.183	27.2	49.0	21.8
	RMU + MCU	46.5	1.205	28.0	48.7	20.7
	RMU + CIR	63.7	1.010	27.9	47.6	19.7
	RMU + CIR + SAM	57.2	1.006	24.2	33.0	8.8
	RMU + CIR + MCU	63.5	1.010	32.2	39.5	7.3
	MLP Breaking	58.7	1.364	29.9	58.8	28.9
	MLP Breaking + SAM	57.5	1.212	21.8	48.5	26.7
	MLP Breaking + MCU	60.3	1.357	29.1	57.5	28.5
	MLP Breaking + CIR	64.8	1.001	22.1	29.7	7.6
	MLP Breaking + CIR + SAM	64.9	1.002	25.9	31.4	5.5
	MLP Breaking + CIR + MCU	64.5	1.001	22.8	26.4	3.6
Years	Original model	65.1	1.000	68.4	-	-
	GA	63.9	1.529	46.1	64.0	18.0
	GA + SAM	56.3	1.360	25.8	63.7	37.9
	NPO	58.5	1.404	27.9	63.4	35.6
	NPO + SAM	56.0	1.363	25.8	62.9	37.1
	RMU	56.8	1.204	33.0	64.3	31.4
	RMU + SAM	54.9	1.201	34.0	65.6	31.6
	RMU + MCU	52.0	1.193	30.8	54.9	24.1
	RMU + CIR	57.1	1.102	30.7	37.4	6.7
	RMU + CIR + SAM	57.1	1.111	29.7	36.6	6.9
	RMU + CIR + MCU	57.2	1.101	31.7	33.7	2.0
	MLP Breaking	60.8	1.239	27.0	51.5	24.5
	MLP Breaking + SAM	58.0	1.215	25.6	47.4	21.8
	MLP Breaking + MCU	61.5	1.214	29.0	49.5	20.6
	MLP Breaking + CIR	64.6	1.010	32.7	39.4	6.7
	MLP Breaking + CIR + SAM	64.3	1.021	26.6	36.9	10.3
	MLP Breaking + CIR + MCU	63.8	1.010	25.9	30.9	5.0

Table 6: Matched operating points for MLP Breaking + CIR and MLP Breaking + CIR + MCU. **Bold** marks the lower Relearn and Δ.

Dataset	Method	Steps	Forget ↓	Relearn ↓	Δ ↓	WikiText
WMDP-Cyber	MLP Breaking + CIR	96	23.6	38.9	15.3	1.0001
	MLP Breaking + CIR + MCU	208	23.9	**32.6**	**8.7**	0.9991
WMDP-Bio	MLP Breaking + CIR	154	23.2	29.6	6.5	1.0007
	MLP Breaking + CIR + MCU	231	22.8	**25.4**	**2.5**	1.0005

Table 7: Robustness against the adaptive representation-based attack on Llama-3.1-8B. The attacker directly aligns the unlearned model's activations with the original model's. **Bold** indicates the lowest Δ (best robustness).

	WMDP-Cyber			WMDP-Bio		
Method	Forget	Relearn	Δ	Forget	Relearn	Δ
Original model	57.6	–	–	64.0	–	–
GA	25.2	58.6	33.4	35.1	64.7	29.7
MLP Breaking + CIR	19.1	35.8	16.6	20.5	31.8	11.3
MLP Breaking + CIR + MCU	25.8	28.7	**2.9**	23.9	28.0	**4.1**

K Cross-Model Generality

To assess whether the benefits of MCU transfer beyond Llama-3.1-8B, we evaluate it on two additional model families, **Gemma2-9B** and **Qwen3-8B**, across all three forget datasets used in the paper. We use the same training and evaluation pipeline as in Appendix H, and compare the strongest representation-based baseline (MLP Breaking + CIR) against the corresponding MCU variant (MLP Breaking + CIR + MCU).

Table 8 reports the results. Across both Gemma2-9B and Qwen3-8B, adding MCU consistently lowers the post-attack relearning gap Δ over the MLP Breaking + CIR baseline while keeping MMLU essentially unchanged. The improvement is substantial on WMDP-Cyber for both models (Gemma2-9B: $4.0 \rightarrow 1.9$; Qwen3-8B: $13.7 \rightarrow 7.4$) and on Years (Gemma2-9B: $6.3 \rightarrow 2.7$; Qwen3-8B: $1.1 \rightarrow 0.8$); on WMDP-Bio with Qwen3-8B, MCU even drives Δ slightly negative, indicating that the relearning attack fails to recover any forgotten knowledge above the post-unlearning level. These results corroborate our main finding that explicitly redirecting forgetting into the minor-component subspace yields more robust unlearning, and that this benefit is not specific to a single base model.

L Consistency of Observations 2–3 Across Unlearning Losses

In Section 3, the change-ratio (Equation 5) and recovery-ratio plots in Figure 2 are reported for GA. Here we show that the same qualitative pattern – unlearning concentrates changes in dominant components, and relearning preferentially recovers them – holds across the full set of unlearning losses considered in our experiments: NPO, RMU, MLP Breaking, and their CIR-augmented variants (RMU + CIR, MLP Breaking + CIR). All experiments use full fine-tuning on Llama-3.1-8B with the WMDP-Cyber forget set, following the setup in Section 3.

Figure 4 reports, for each method, (left) the unlearn change ratio across principal-component indices and (right) the corresponding recovery ratio after the RTT relearning attack. Across all five methods, the change-ratio mass is concentrated in the first few PCs, and the recovery ratio is high precisely for these dominant components and decays toward the minor components. This confirms that Observations 2 and 3 are not specific to GA, but reflect a property shared by representation-level (RMU, MLP Breaking) and output-level (NPO) unlearning losses, with and without CIR-style gradient filtering. Equivalently, the dominant-component vulnerability that motivates MCU is a generic property of current LLM unlearning pipelines rather than an artifact of any particular loss.

Table 8: Cross-model evaluation on Gemma2-9B and Qwen3-8B across all three datasets. MCU is applied on top of the best MLP Breaking + CIR setting. Lower Δ is better.

Model	Dataset	Method	MMLU	Forget	Relearn	Δ
Gemma2-9B	WMDP-Cyber	MLP Breaking + CIR	69.8	26.3	30.3	4.0
		MLP Breaking + CIR + MCU	69.9	22.4	24.3	**1.9**
	WMDP-Bio	MLP Breaking + CIR	69.7	27.8	33.3	5.5
		MLP Breaking + CIR + MCU	69.4	23.0	28.1	**5.1**
	Years	MLP Breaking + CIR	69.6	30.0	36.3	6.3
		MLP Breaking + CIR + MCU	68.9	26.2	28.9	**2.7**
Qwen3-8B	WMDP-Cyber	MLP Breaking + CIR	75.7	28.0	41.7	13.7
		MLP Breaking + CIR + MCU	75.8	29.0	36.4	**7.4**
	WMDP-Bio	MLP Breaking + CIR	73.4	26.6	35.5	8.9
		MLP Breaking + CIR + MCU	75.1	29.5	26.1	**-3.4**
	Years	MLP Breaking + CIR	75.4	39.4	40.6	1.1
		MLP Breaking + CIR + MCU	75.3	30.1	30.9	**0.8**

M Robustness of Observations to Forget-Set Size

A natural concern is whether Observations 1–3 are artifacts of a particular forget-set size, or could be driven by the size of the sample used to fit PCA. To rule this out, we run two complementary experiments on WMDP-Cyber with GA and Llama-3.1-8B (full fine-tuning), varying the forget-set fraction in $\{25\%, 50\%, 75\%, 100\%\}$.

Experiment A: fixed model, varying PCA-fit subset. We keep the unlearned and relearned models fixed (trained on the full forget set) and only vary the size of the subset used to fit PCA. This isolates whether Observation 1 (variance concentration in the dominant components) is a consequence of using too few samples for PCA. Figure 6 shows the explained-variance curves: the spectrum is essentially indistinguishable across 25%, 50%, 75%, and 100% subsets, so variance concentration is not a sample-size artifact. Correspondingly, Figure 7 reports the change-ratio and recovery-ratio plots for the four subset sizes; both retain the same dominant-component-heavy pattern.

Experiment B: end-to-end varying forget size. We then re-run the full unlearning + relearning pipeline on each forget-set fraction, so both the model and the PCA fit are tied to the same subset. This tests whether the observations hold when the unlearning procedure itself is varied. Figure 8 shows that the change-ratio and recovery-ratio patterns remain consistent across all subset sizes: a small number of dominant PCs continue to absorb the bulk of unlearning-induced change and to be preferentially recovered after RTT.

Together, Experiments A and B confirm that all three observations are robust to the size of the forget set, and that the dominant-component vulnerability is a structural property of LLM representations rather than a consequence of a specific sampling regime.

N Robustness of Observations to Parameter-Efficient Fine-Tuning

Our main analysis in Section 3 uses full fine-tuning. To rule out the possibility that Observations 2–3 are an artifact of full-parameter optimization, we additionally evaluate them under **LoRA** fine-tuning at ranks $r \in \{8, 16, 32, 64\}$, applied to four unlearning losses (GA, NPO, RMU, MLP Breaking). Observation 1 concerns the representation geometry of the pre-trained model itself and is therefore independent of the fine-tuning method, so we focus on Observations 2 and 3.

Figures 9 to 12 report, for each unlearning loss, the per-PC unlearn change ratio (left) and recovery ratio (right) under Full FT and the four LoRA ranks. In every row, the dominant components again absorb the majority of the unlearning-induced change and exhibit the highest recovery, matching

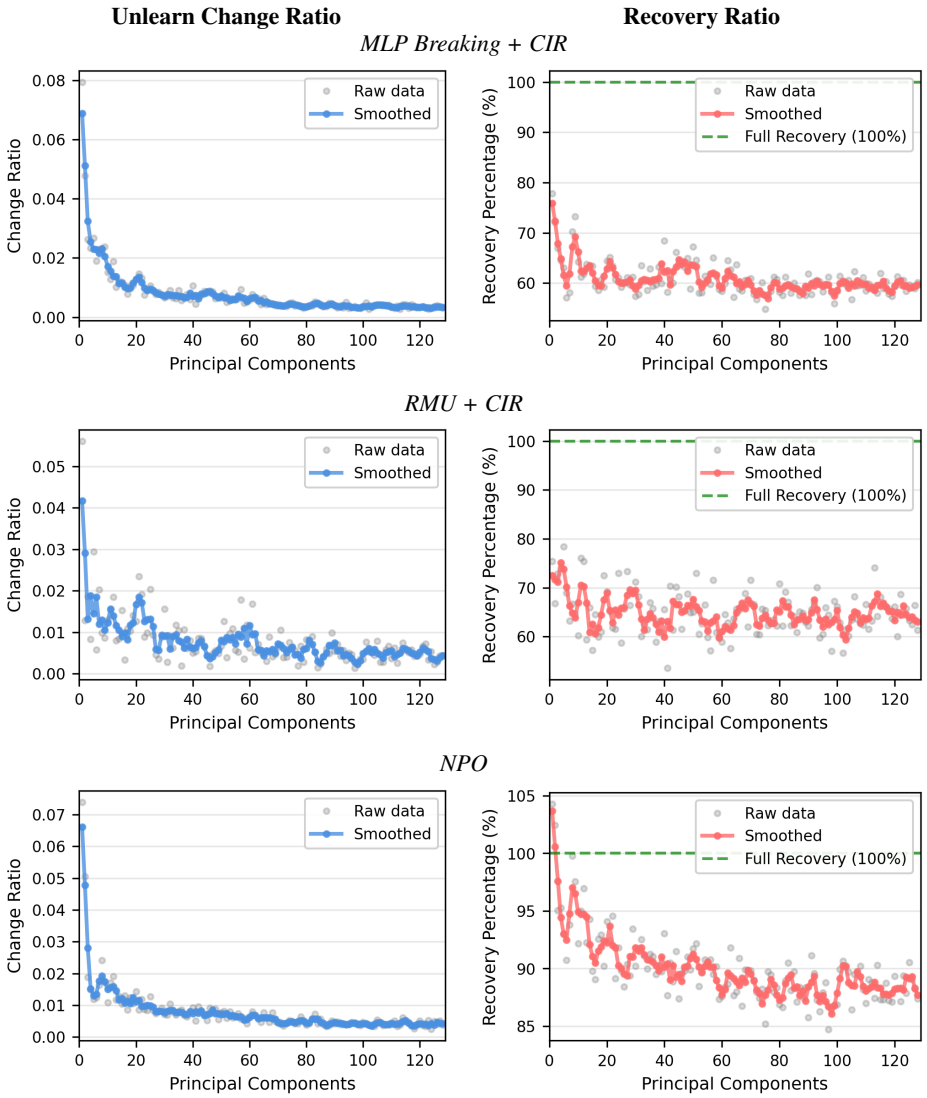


Figure 4: Consistency of Observations 2–3 across unlearning losses on WMDP-Cyber (Llama-3.1-8B, full fine-tuning), part 1. For each method, the left plot shows the per-PC change ratio induced by unlearning and the right plot shows the per-PC recovery ratio after the RTT relearning attack.

the full fine-tuning pattern in Figure 2. This holds uniformly across the four unlearning losses and the four LoRA ranks, confirming that the dominant-component vulnerability is a property of LLM representation geometry rather than of the specific optimization regime, and that MCU's design (targeting the minor-component subspace) is therefore equally motivated for LoRA-based unlearning pipelines.

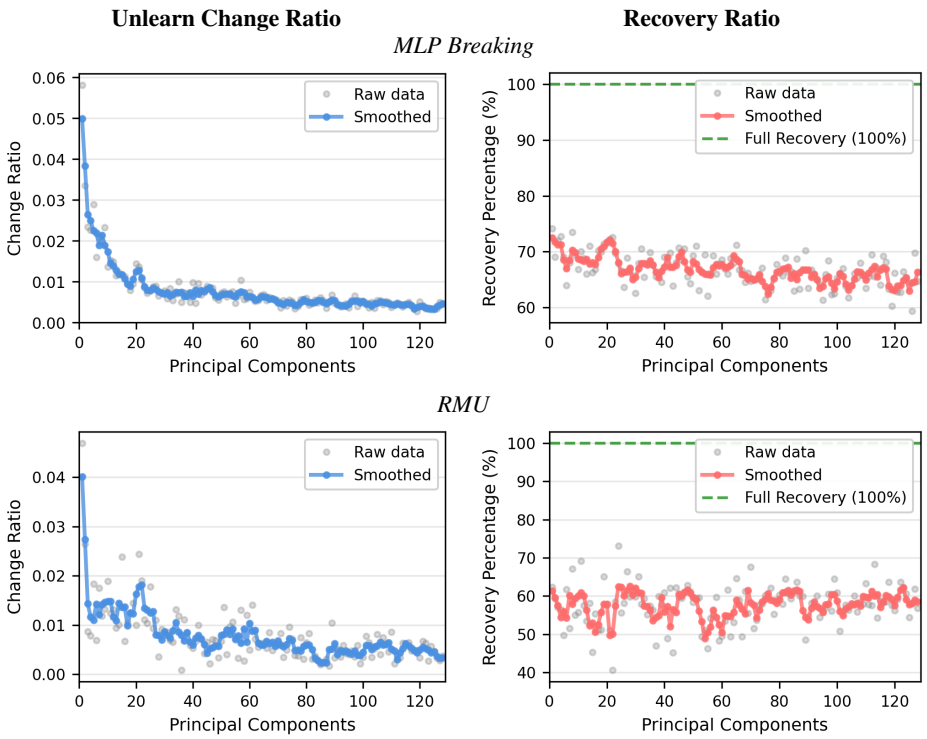


Figure 5: Consistency of Observations 2–3 across unlearning losses on WMDP-Cyber (Llama-3.1-8B, full fine-tuning), part 2. The dominant components consistently absorb most of the unlearning change and exhibit the highest recovery, regardless of the specific unlearning loss.

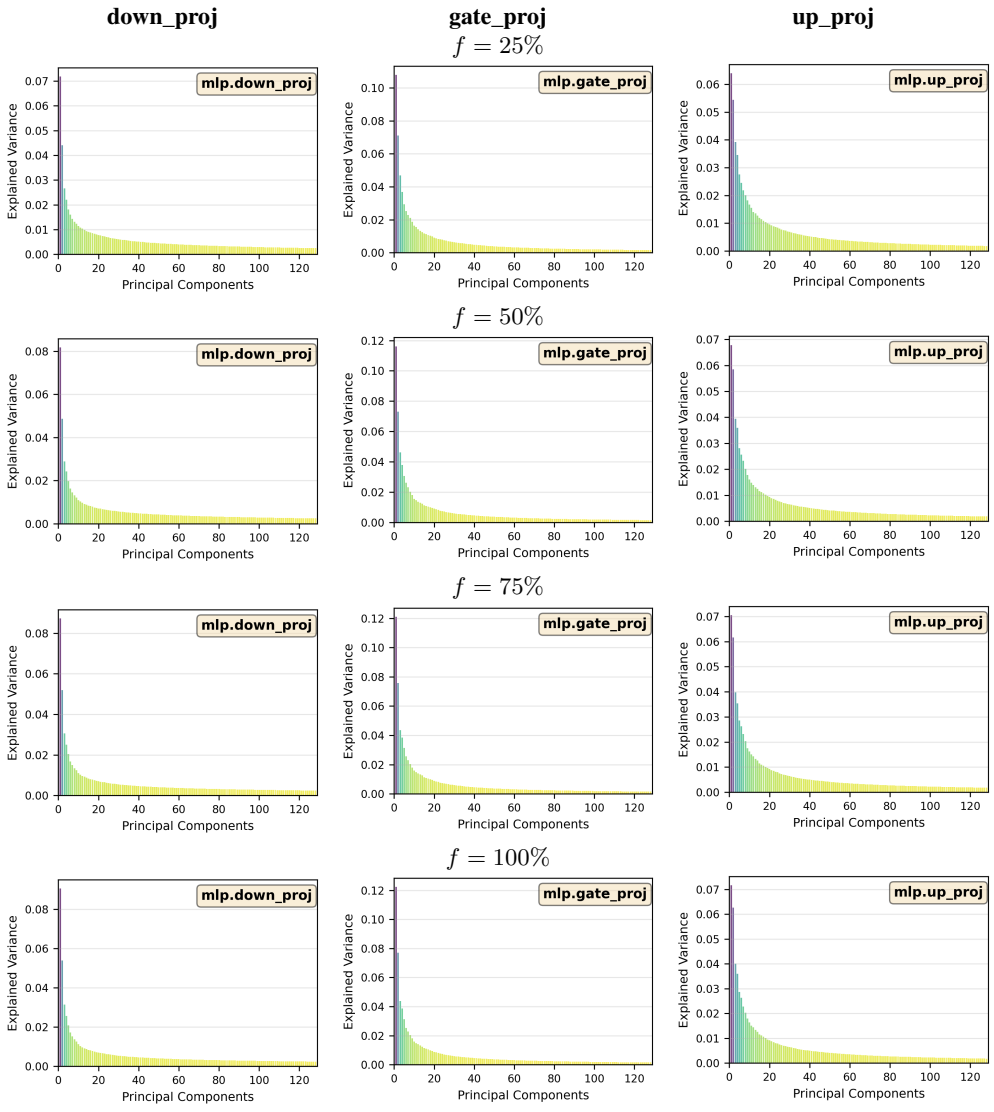


Figure 6: Experiment A: explained variance under varying PCA-fit subset sizes (fixed unlearned/relearned model). The spectrum is essentially identical across 25–100% subsets, so variance concentration is not driven by sample size.

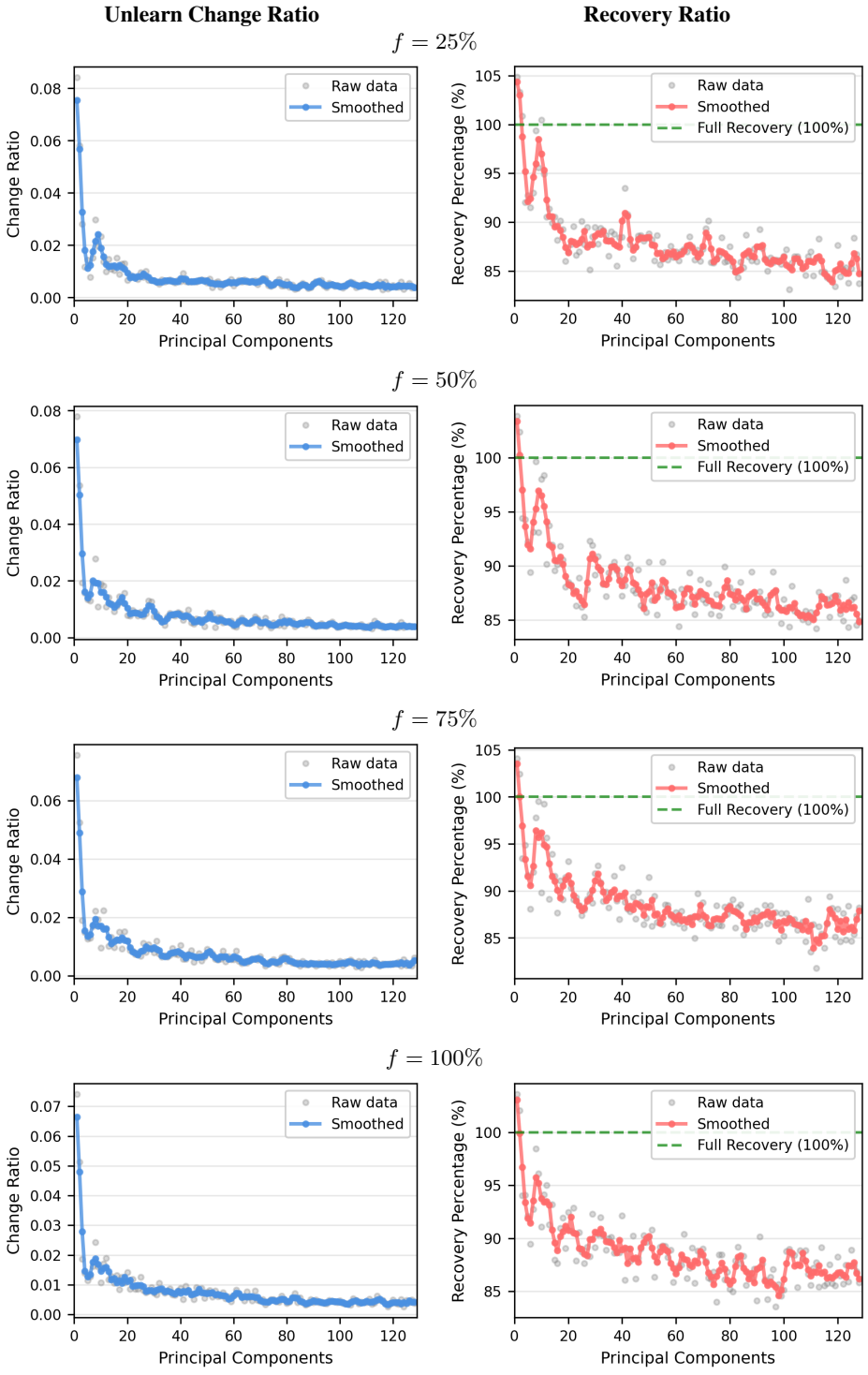


Figure 7: Experiment A: per-PC unlearn change ratio (left) and recovery ratio (right) as a function of the PCA-fit subset size (rows: $f = 25, 50, 75, 100\%$), with the unlearned/relearned model held fixed. Observations 2 and 3 are stable across PCA-fit sizes.

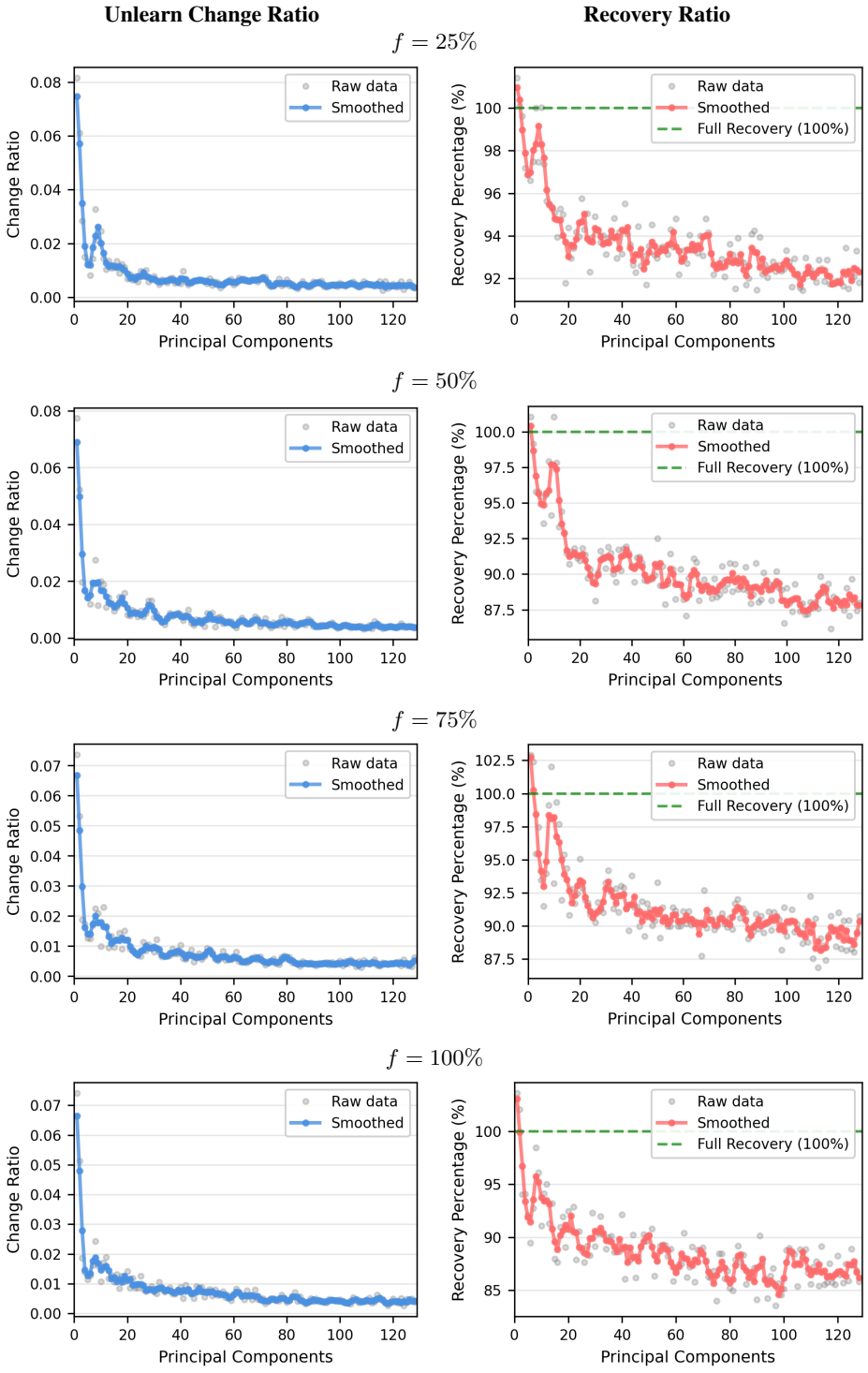


Figure 8: Experiment B: per-PC unlearn change ratio (left) and recovery ratio (right) under end-to-end varying forget size (rows: $f = 25, 50, 75, 100\%$). The unlearning + relearning pipeline is rerun on each subset. Observations 2 and 3 hold across all forget-set sizes.

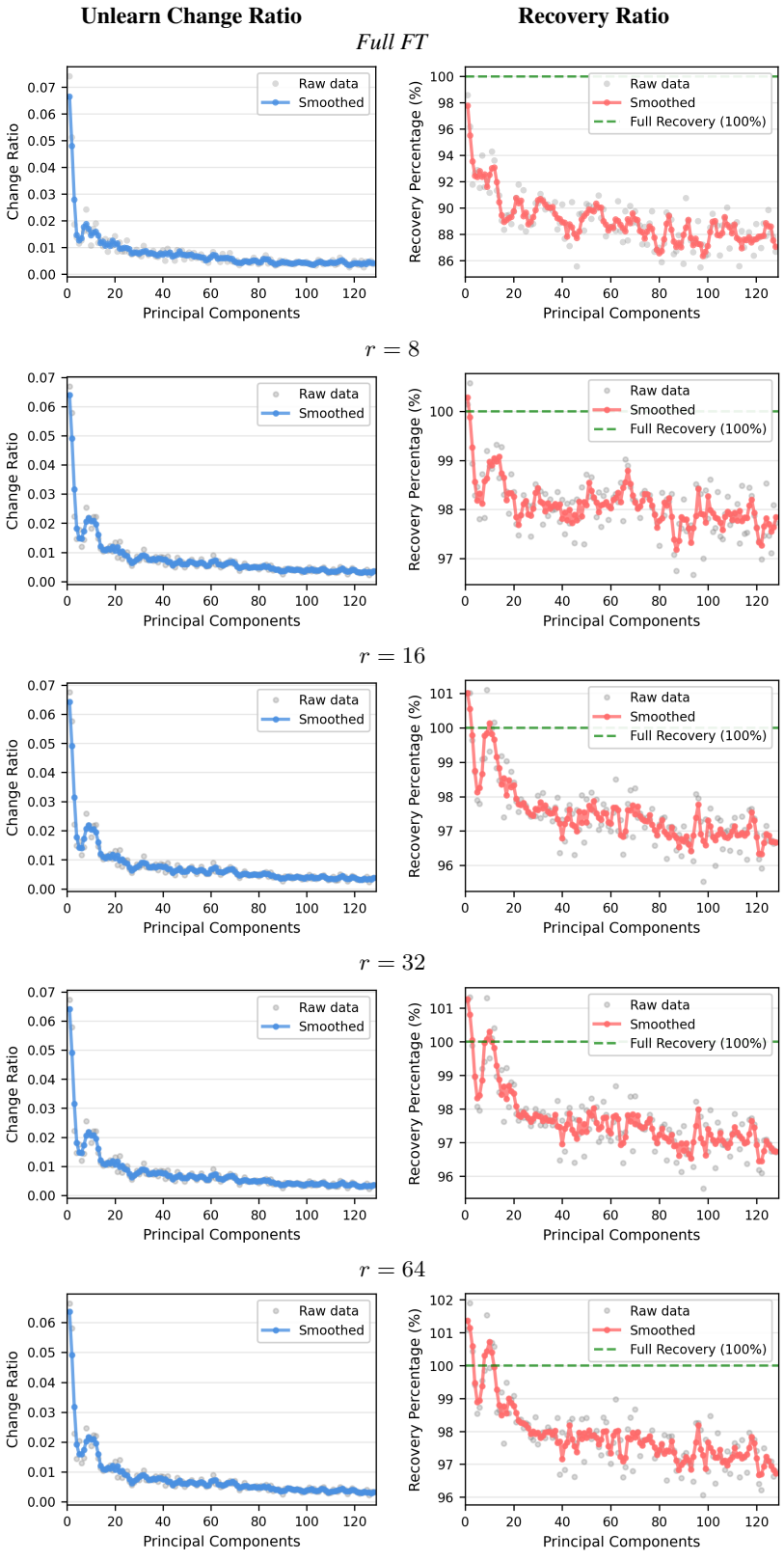


Figure 9: GradDiff under Full FT and LoRA at four ranks. Per-PC unlearn change ratio (left) and recovery ratio (right) on WMDP-Cyber (Llama-3.1-8B). Dominant components dominate both change and recovery for every optimization regime.

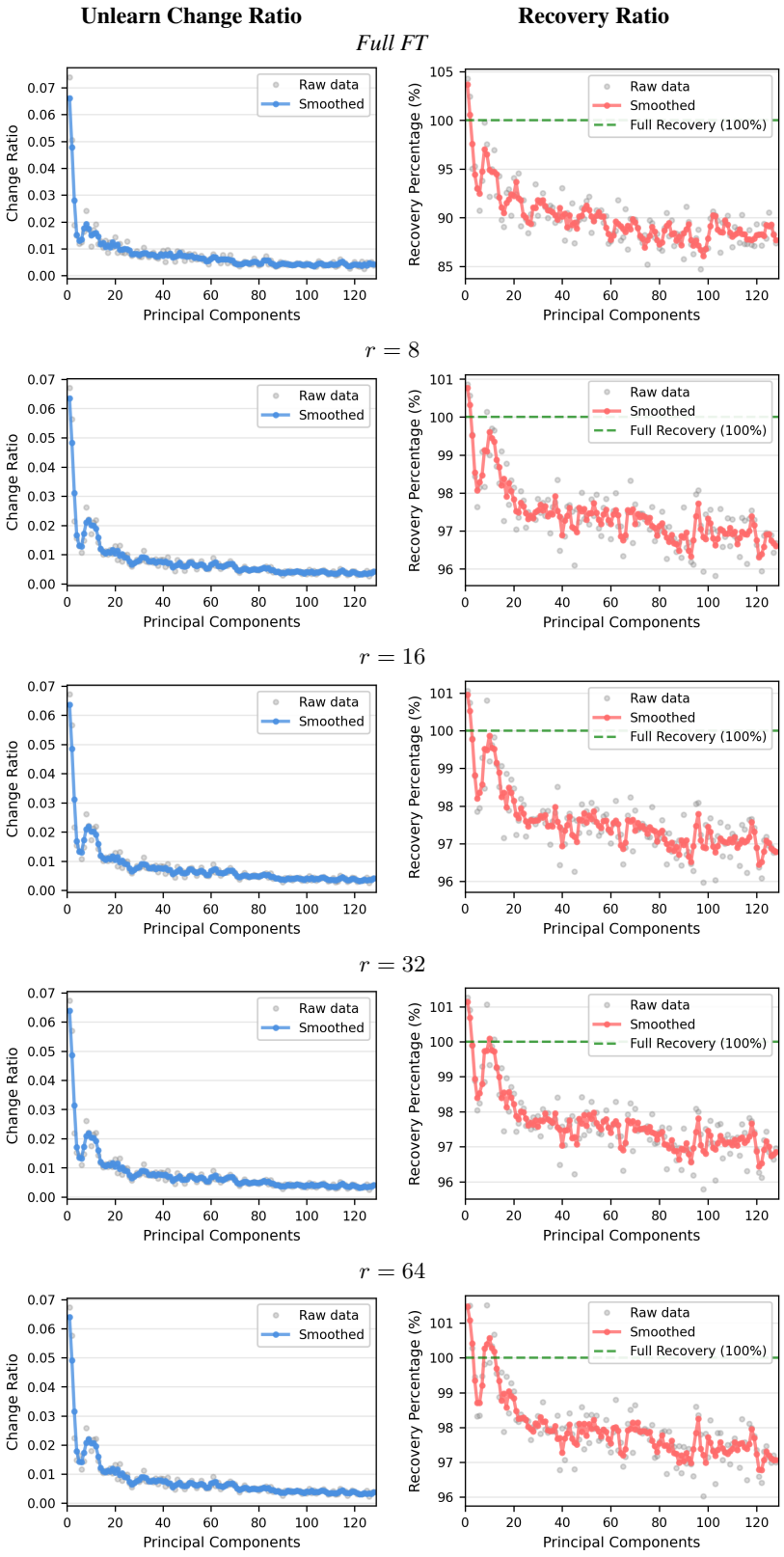


Figure 10: NPO under Full FT and LoRA at four ranks. Same layout as Figure 9.

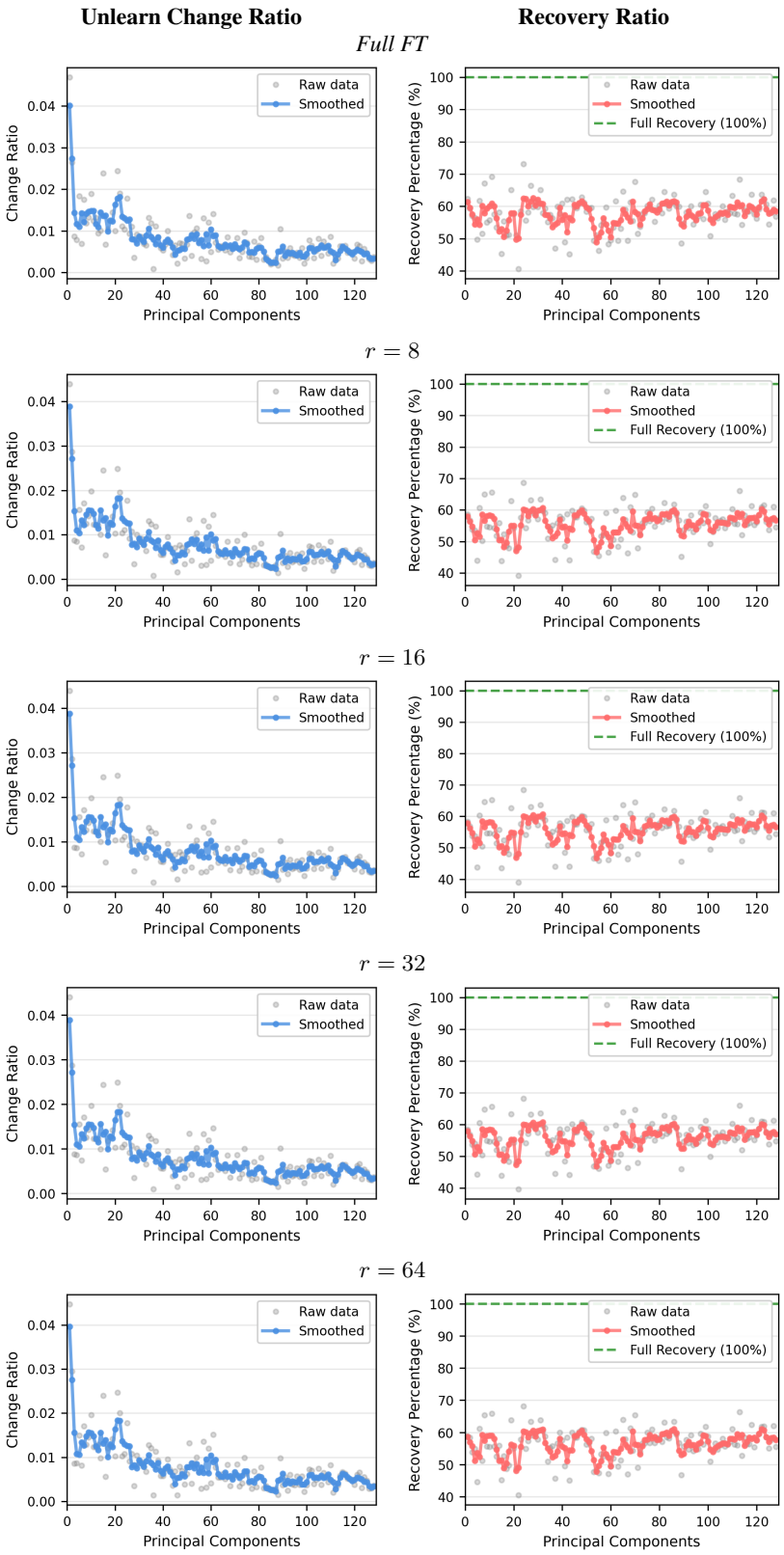


Figure 11: RMU under Full FT and LoRA at four ranks. Same layout as Figure 9.

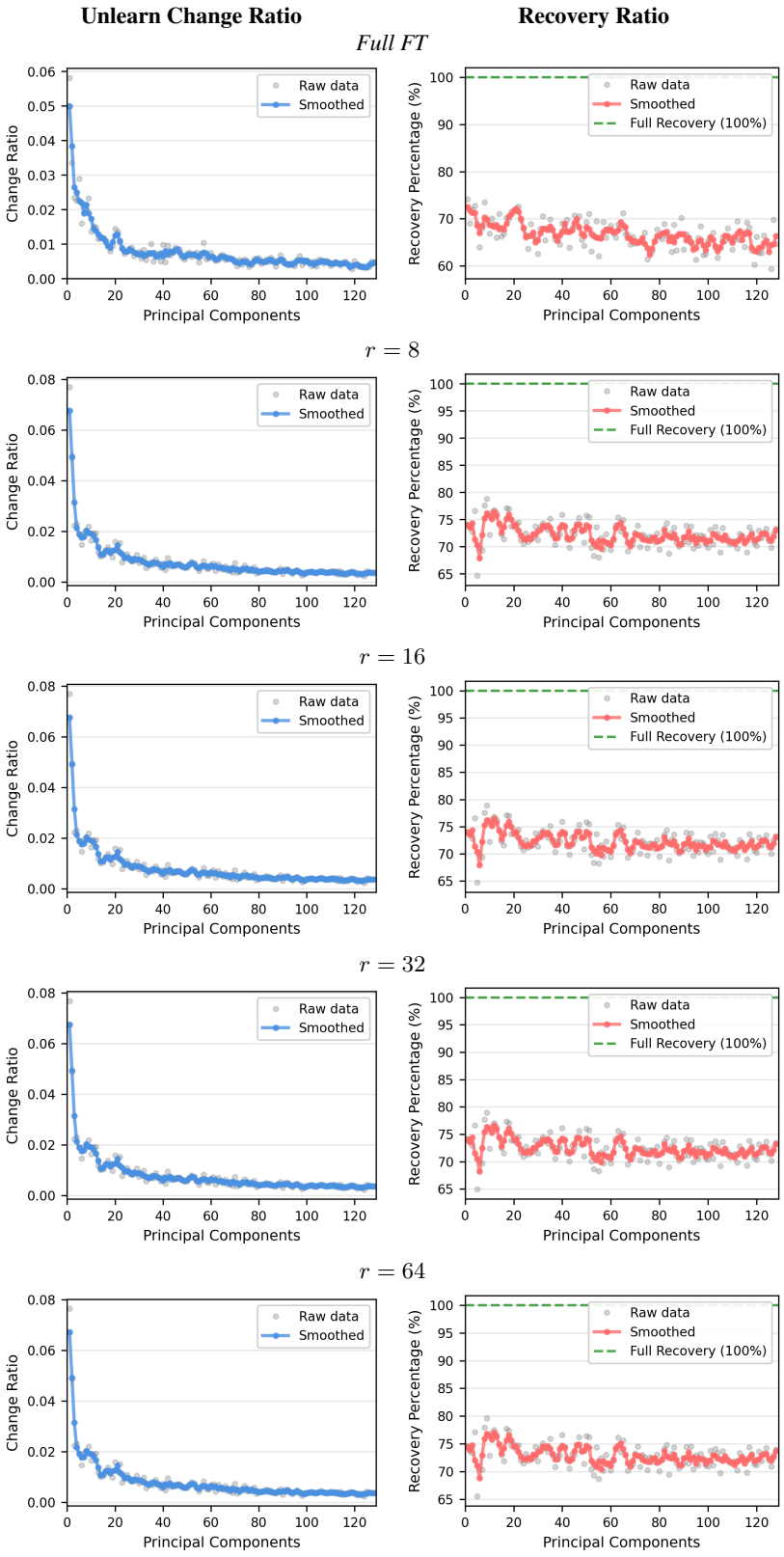


Figure 12: MLP Breaking under Full FT and LoRA at four ranks. Same layout as Figure 9.