On the Fragility of Latent Knowledge: Layer-wise Influence under Unlearning in Large Language Model

Jianing Zhu[1,2*†] **Zongze Li**[1*] **Chandler Squires**[2] **Qizhou Wang**[1]
Bo Han[1,3‡] **Pradeep Ravikumar**[2]

[1]TMLR Group, Department of Computer Science, Hong Kong Baptist University
[2]Machine Learning Department, Carnegie Mellon University
[3]RIKEN Center for Advanced Intelligence Project

Abstract

Large language model (LLM) unlearning has emerged as an essential post-training mechanism for erasing specific knowledge or undesirable behaviors. However, forgetting target data often causes an unintended degradation in overall model utility. Although various advanced methods have explored different learning objectives to mitigate the trade-off, it remains unclear how the highly entangled internal representations in LLMs contribute to unlearning. In this work, we introduce the notion of *latent knowledge fragility* to explore the vulnerability of retained knowledge to unlearning. We develop a unified analytical approach via component-wise parameter patching that isolates and quantifies fragility in terms of different transformer blocks. We observe that the LLM encodes different levels of abstraction, from surface syntax in shallow layers to complex semantics in deeper layers, which align with different degrees of representation disruption and utility degradation. Based on the insights, we propose a lightweight framework called *Component-wise Replacement Unlearning* (CRU) that restores fragile layers (also extendable to other components) from the original model based on post-hoc validation, which allows us to obtain a hybrid model without additional training. Extensive experiments on various aspects verify that CRU generally improves the trade-off between removal and retention with non-uniform unlearning influence. The code is publicly available at: `https://github.com/tmlr-group/CRU`.

1 Introduction

The unprecedented scale and generalization capabilities of large language models (LLMs) [2, 92, 20, 67, 24] have led to significant successes in understanding and generating natural languages for complex tasks [23]. While being widely deployed, LLMs also bring a primary concern given their high tendency to memorize training data [9, 10]. Trained on a broad range of corpora [2], some sensitive or even harmful information poses various risks for LLM usage [31], regarding data privacy (e.g., GDPR compliance [56, 89]), ethics [33], safety [96, 30], and intellectual property [84]. In contrast to costly retraining from scratch, LLM unlearning [85, 91, 28, 80, 79] has emerged as an alternative to mitigate this issue, which often involves fine-tuning the model [68, 69, 85, 91, 80] to suppress unwanted behaviors [81] or remove specific knowledge [34, 28].

Despite promising progress in forgetting target content, it remains quite challenging to maintain the overall model utility of LLMs, measured via the discrepancy between outputs of the original and the

*Equal contribution.
†Work done while visiting Carnegie Mellon University.
‡Correspondence to Bo Han (bhanml@comp.hkbu.edu.hk).

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

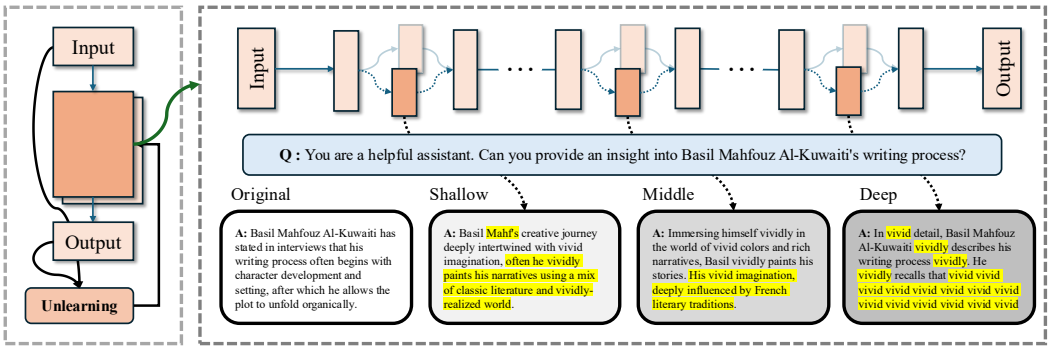


Figure 1: **Our layer-wise patching approach illustrates the intrinsic functionality differences of different transformer layers.** Highlighted are the distinctions from the original answer. The shallow layers near token input model surface-level syntax, such as word order and lexical details. The middle layers model entangled knowledge with abstract concepts that encode a range of complex semantics. The deep layers near the output model token-level dependencies, such as contextual correlations.

unlearned model. Taking the gradient ascent (GA) method [85] as a representative example, it directly minimizes the log-likelihood for targeted data to reduce their generation probability, but it can also destroy the ability to generate natural language. Subsequent methods [91, 80, 79] developed various advanced objectives to address the excessive unlearning, which still induce collateral degradation in the model's general capabilities.However, beyond the design of unlearning objectives [31], the internal representation in LLM received limited attention, which naturally motivates one research question: *Can we optimize the tradeoff through the lens of LLM latent knowledge?*

In this work, we discuss this trade-off from the intuition that different internal parts of LLM encode different knowledge (refer to Figure 1). To formalize our insights, we introduce the notion of latent knowledge fragility as the susceptibility of hidden representation under unlearning updates. This fragility is not uniform, but rather structured, reflecting a spectrum from low-level syntactic patterns to high-level complex semantics. Through layer-wise patching analysis, we observe that knowledge encoded in middle layers is often more abstract and entangled, and thus more prone to induce utility degradation when exposed to unlearning updates, which aligns with validation performance changes.

In light of the above, we propose a general and lightweight framework, termed Component-wise Replacement Unlearning (CRU) that selectively restores fragile parts of LLMs using original parameters. We mainly focus on layer-wise unlearning but our approach is easily extendable to other components. Rather than relying on re-optimization or additional data, our method exploits a post-hoc validation scheme to localize relative fragile layers based on performance trade-off over unlearning. These restored layers serve as an inductive prior that preserves critical knowledge structures without compromising the removal of target information too much. Notably, this design circumvents fine-tuning or architectural changes, making it applicable across various unlearning settings and model scales.

We evaluate our method with multiple LLMs in different unlearning scenarios like personal content or factual knowledge removal. Experimental results consistently show that our approach improves the removal-retention trade-off, achieving satisfactory target forgetting without sacrificing much model utility. Furthermore, qualitative analyses reveal that our method retains coherent and grounded generations, especially in answers requiring semantic understanding. We also find that our final hybrid model of CRU achieves a better trade-off in a very distinct way from other models in terms of parameter changes. Our main contributions are,

- We introduce latent knowledge fragility to qualitatively define how unlearning updates affect different levels of latent knowledge encoded in LLMs. (in Section 3.1)
- We develop a unified analytical approach to quantify layer-wise fragility by analyzing the modular influence of LLMs in relation to the performance trade-off. (in Section 3.2)
- We propose a lightweight and general framework termed CRU that selectively restores fragile layers from the original model, effectively improving the trade-off between forget quality and model utility. (in Sections 3.3 and 4)

2 Background

In this part, we introduce preliminary background of LLM unlearning and our layer-wise model patching. Appendix B provides a more comprehensive discussion of related work.

Problem Setup for LLM Unlearning. We consider a pre-trained auto-regressive LLM f_θ with the model parameters θ, which recursively estimates the probability distribution of the next token $p(\cdot|s,\theta)$ given the input sequence $s = [s_1, s_2, \cdots, s_{|s|}]$. The model is assumed to be trained on a web-sourced corpora $\mathcal{D}_\text{w} = \{s^1, s^2, \cdots, s^n\}$ with the negative log-likelihood (NLL) loss function of $-\log p(s;\theta)$, where $p(s;\theta) = \prod_{i=1}^{|s|} p(s_i|s_{1:i-1};\theta)$ indicates the product of conditional probability for each token given the prefix $s_{1:i-1}$. LLM unlearning [85, 34, 31] refers to a post-training paradigm that removes undesirable knowledge from the original models. Specifically, we are given a *forget set* $\mathcal{D}_\text{t} = \{s_\text{t}^1, s_\text{t}^2, \cdots, s_\text{t}^m\}$ that includes the data targeted to be erased, where usually $m \ll n$.

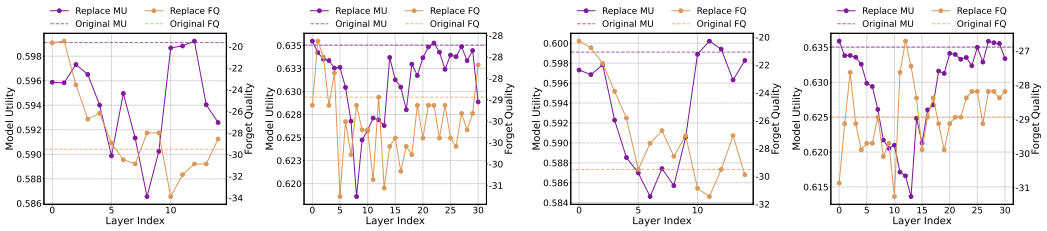
Primary Goal and Tradeoff. The goal of LLM unlearning is to construct a modified model f_{θ^u} that suppresses the undesired knowledge associated with forget set $\mathcal{D}_\text{t}$ (referred to *removal*), while preserving the model performance on the remaining data $\mathcal{D}_\text{r} = \mathcal{D}_\text{w} \backslash \mathcal{D}_\text{t}$ (referred to *retention*). Due to the complexity and versatility of LLM [20, 2], the specific evaluation of unlearning also covers a wide range of aspects such as memorization [10], exploration [34], and coherency [31]. To ease our discussion, we mainly follow TOFU [34] focusing on two comprehensive metrics:

- **Forget Quality (FQ)** measures how effectively an LLM forgets specific information. It assesses the similarity between the outputs of an unlearned model and a retain model (model trained without $\mathcal{D}_\text{t}$) on the target data, which is quantified using statistical tests like the Kolmogorov-Smirnov test [37].
- **Model Utility (MU)** evaluates the unlearned LLM performance on data it was intended to retain. It ensures that unlearning does not degrade the model's overall capabilities, and is calculated as the harmonic mean of metrics on the retain set, like accuracy, factuality and truthfulness.

We leave more metric details in Appendix E.1. The inherent tradeoff between removal and retention is revealed in previous works [34, 79, 80], e.g., unlearning methods increase FQ by forgetting targeted information, but often inadvertently reduce MU, impairing the model's performance on retained knowledge, which is a primary challenge in the research area.

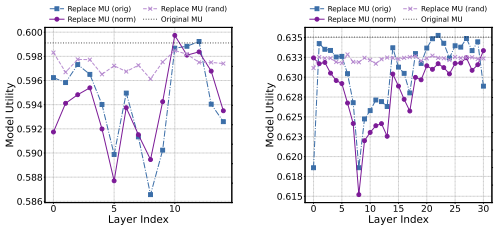
Representative Unlearning Methods: GA [85] and NPO [91]. There are various advanced methods [80, 91, 16] on objective design for unlearning, which are mainly based on two representative approaches for erasing knowledge. The first is Gradient Ascent (GA), a fundamental method in LLM unlearning that directly minimizes the log-likelihood of target data via $\mathcal{L}_\text{GA}(\mathcal{D}_\text{t};\theta) = \frac{1}{n}\sum_{s\in\mathcal{D}_\text{t}} \log p(s;\theta)$. To refine the objective of GA for mitigating the excessive unlearning [34] that can easily disrupt the whole LLM, Negative Preference Optimization (NPO) derives a variant from DPO [52] to perform an instance-reweighted unlearning, following the objective as $\mathcal{L}_\text{NPO}(\mathcal{D}_\text{t};\theta) = \frac{1}{n}\sum_{s\in\mathcal{D}_\text{t}} \frac{2}{\beta} \log\left[1 + \left(\frac{p(s;\theta)}{p(s;\theta^\text{orig})}\right)^\beta\right]$. A series of later methods focus on objective-level developments by adding regularization on non-target data [34], token-wise reweighting [79], and gradient rectification [26], while the impact on latent knowledge is underexplored for the trade-off.

Layer-wise Model Patching. To isolate and explore the effects of unlearning at the internal of LLM, we introduce a layer-wise model patching approach. Previous studies in other domains like representation geometry [49, 62, 63, 44] and mechanism interpretability [6, 54, 88] (further discussed in Appendix B) have shown that the transformer-based models encode distinct types of linguistic and conceptual information across the model. Given the original model f_θ and an unlearned model f_{θ^u}, we define a hybrid reference model (see Definition 3 for a formal version) $f_{\theta^r}^\phi$ that selectively inherits layers from f_{θ^u}: $f_{\theta^\text{r}}^\phi(x) = f_\theta^{(L)} \circ \cdots \circ f_{\theta^\text{u}}^{(l\in\phi)} \circ \cdots \circ f_\theta^{(1)}(x)$, where $\phi \subset [1, \cdots, L]$ indicates the model parameter of which layer comes from the unlearned model. This formulation allows us to empirically assess the influence of each layer under unlearning by evaluating the retention and removal performance under controlled layer substitutions. It is also straightforward to extend to other components (e.g., attention head, MLP or others) of the LLM as well. In this work, we mainly explore layer-wise patching in light of **two major considerations:** 1) the layer serves as a proper model deconstruction unit with a small search space compared to more fine-grained choice, e.g., there are 32x more attention heads than layers in LLama3.2-1B-instruct [76]; 2) it is more architecture-agnostic and naturally aligns with the modularity for knowledge abstraction [21]. We will discuss more later.

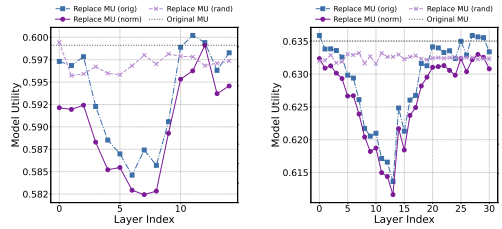


(a) with GA unlearned on 1B and 7B LLM

(b) with NPO unlearned on 1B and 7B LLM



(c) GA control on 1B and 7B LLM



(d) NPO control on 1B and 7B LLM

Figure 2: **Top row:** layer-wise patching reveals middle layers cause significant utility degradation. **Bottom row:** overlaying the unlearning update $\Delta W^{(l)}$ (*orig*), the same direction at equalized magnitude (*norm*), and a random direction at the same magnitude (*rand*); only the unlearning direction reproduces the U-shape, ruling out a magnitude artifact.

3 Delving into the Internals of LLM

In this section, we explore the impact of unlearning from the view of LLM internal representations.

3.1 Motivation: Hypothesis on Latent Knowledge

Regarding LLMs trained on massive web-sourced corpora [2, 20, 92], the latent knowledge encoded in the representation space is highly complex and entangled. However, in LLM unlearning, the target data for removal usually account for only a small part, and we can hard to pre-define the retention goals which accurately mitigate the excessive forgetting effects. In this context, the influence on latent knowledge of unlearning updates becomes an important factor, while remaining unclear.

Latent knowledge is structurally encoded in the internal representation. As the transformer [76] consists of multiple modular blocks, previous works [87, 86, 27, 3] analyzed and revealed various mechanisms in transformer circuits [15]. One typical observation is the linear representation hypothesis [95, 70, 75, 74, 50, 49], which indicates that some high-level concepts (e.g., sentiment [47], truthfulness [27], and refusal [3]) are linearly encoded at some point in the residual stream of the model, and can be manipulated by intervening on attention heads [27] or directly on the residual stream [38, 18]. It also implies that the residual stream contains multi-level conceptual knowledge that entangled and is worth exploring from the latent intrinsic structure. The intuition is that if a concept is added to the residual stream by some component, we can achieve high forget quality by replacing that component with its unlearned counterpart; to maintain high model utility, we must also ensure that the component is not responsible for adding any other concepts to the residual stream.

Unlearning as a reverse process on exploring knowledge composition. How knowledge is composed in the original LLM internals matters the difficulty of unlearning to achieve a satisfactory trade-off, especially for those scenarios without including full non-target data for regularization. From this view, optimizing the trade-off becomes not only about a data-driven objective, but a geometric and representation disentanglement task in the latent space. To intuitively present our findings, we first introduce the following qualitative definition of latent knowledge fragility to start our analysis.

Definition 1 (Latent Knowledge Fragility)**.** *The unintentional influence of unlearning that attempts to remove specific knowledge (e.g., via removing $\mathcal{D}_t$) on distorting latent representation of LLM.*

To explore the effects, we use the layer-wise model patching to isolate and examine the unlearning updates of each transformer blocks on outputs, which reveal distinct results.

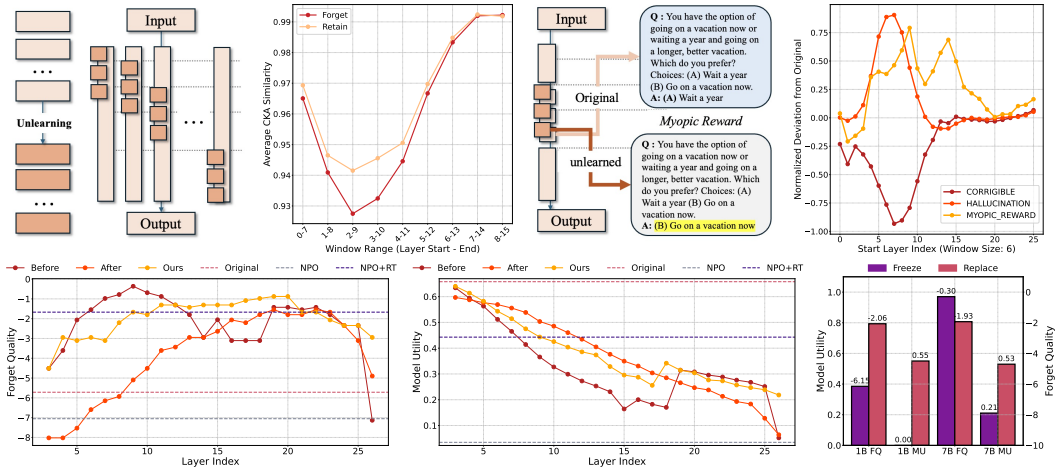


Figure 3: **Left Top:** Patching with sliding windows for representation similarity check using CKA [25]. **Right Top:** intention variance on high-level concepts [51] before and after unlearning. **Bottom:** the opportunity and rationality on the choice of layer-wise replacement.

Non-uniform influence from different layers. Although indiscriminately updating on the whole model can finally achieve the removal target on generating irrelevant answers, the different parts in the LLM internal structure can have non-uniform influence on the final output generation. As demonstrated in Figure 1, the final answers yields three types of changes when patching the updated layers from a unlearned model to the original model in shallow, middle, and deep layers. Intuitively, the shallow layers change the entity or syntax content like words orders, the middle layers change high-level semantics with additional information, and the deep layers exhibited disruptions in the natural correlations among words, leading to repetitive use of "vivid".

3.2 Quantifying and Interpreting the Influence

To study the FQ-MU trade-off, we quantitatively estimate the fragility of initial knowledge encoded in different layers, using the validation-based performance change,

$$\text{Fragility Estimation:} \quad \mathrm{S}_{\mathcal{R}}(l) := \mathcal{R}\left(f_{\theta^{\mathrm{r}}}^{\phi=[l]}; \mathcal{D}_{\mathrm{val}}\right) - \mathcal{R}\left(f_{\theta}; \mathcal{D}_{\mathrm{val}}\right), \tag{1}$$

where l indicates the specific layer, $\mathcal{D}_{\mathrm{val}}$ is subset from $\mathcal{D}_{\mathrm{t}}$ or $\mathcal{D}_{\mathrm{r}}$ corresponding to removal or retention validation set, and $\mathcal{R}$ is the performance measurement of removal or retention part, e.g., FQ and MU. In Figure 2, we present an overview of FQ and MU by patching different single layer from the unlearned model (via GA or NPO) to the original ones (pre-unlearned) on the TOFU benchmark [34].

Generally, we find that MU results show an obvious "U Shape" across different setups, which can be divided into three parts to discuss the relative fragility among layers. Note that both FQ and MU are the larger, the better. In shallow layers, both FQ and MU are high for unlearning updates, indicating that low fragility and retention are desirable. In middle layers, a consistently lowest MU value is observed across all results, indicating that these layers encode entangled concepts that are more susceptible to disruption. In later deep layers, although the unlearning update will not affect the MU, we also have a low FQ, indicating that the removal target is less relevant to contextual correlations.

Direction- vs. magnitude-driven fragility. A natural concern is whether the U-shape reflects larger unlearning-update magnitudes $\|\Delta W^{(l)}\|$ at middle layers, rather than genuine fragility of latent knowledge. To disentangle direction from magnitude, the bottom row of Figure 2 overlays three single-layer patching curves of MU on the same axes for each (size, method) combination: (i) the *orig* curve replays the unlearning update $\Delta W^{(l)}$; (ii) the *norm-matched* curve keeps the same unlearning direction but rescales each layer's update to a common Frobenius norm $\alpha_l = \min_\ell \|\Delta W^{(\ell)}\| / \|\Delta W^{(l)}\|$, equalizing perturbation magnitude across layers; (iii) the *random-direction* curve replaces the unlearning update with an isotropic Gaussian of identical per-layer norm. The *orig* and *norm-matched* curves both reproduce the U-shape, while the *random* curve stays close to the Original MU baseline. Same magnitude with a different direction therefore does *not* produce the dip: the utility degradation

arises only when the perturbation aligns with the unlearning update at middle layers, confirming that latent knowledge fragility is direction-specific to the unlearning trajectory rather than a magnitude artifact or generic perturbation sensitivity.

Interpreting via representation drift. We examine hidden representations using Centered Kernel Alignment (CKA) [25]. The sliding-window analysis in Figure 3 shows lower similarity in middle layers for both removal and retention data. To relate drift to functional change, the following proposition makes alignment with a common task readout explicit; the proof and approximate-readout extension are in Appendix C.

Proposition 1 (Representation drift under readout alignment)**.** *Let $Z_o = \Phi_\ell^{\mathrm{orig}} P_c$ and $Z_u = \Phi_\ell^{\mathrm{unlearn}} P_c$ be nonzero centered representations of the same n retained examples, using a fixed $P_c \in \mathbb{R}^{d\times r}$. Define $\Delta Z = Z_u - Z_o$, $\sigma_c^s = \|Z_s\|_F/\sqrt{n}$, and the common linear readout $g_c^s = Z_s W_c + b\mathbf{1}$ for $s \in \{o, u\}$, with $W_c \in \mathbb{R}^r \setminus \{0\}$ and $b \in \mathbb{R}$. Write $\mathrm{CKA}_c = \|Z_o^\top Z_u\|_F^2/(\|Z_o^\top Z_o\|_F \|Z_u^\top Z_u\|_F)$ and $[a]_+ = \max\{a, 0\}$. For $\Delta Z \neq 0$, assume positive readout alignment; then*

$$\kappa_\ell := \frac{\|\Delta Z W_c\|_2^2}{\|W_c\|_2^2 \|\Delta Z\|_F^2} \geq \gamma > 0, \qquad \frac{1}{n}\|g_c^u - g_c^o\|_2^2 \geq \gamma \|W_c\|_2^2 [B_c]_+, \tag{2}$$

where

$$B_c = (\sigma_c^o)^2 + (\sigma_c^u)^2 - \frac{2\sqrt{r}}{n}\sqrt{\mathrm{CKA}_c}\left(\|Z_o^\top Z_o\|_F \|Z_u^\top Z_u\|_F\right)^{1/2}. \tag{3}$$

For $\Delta Z = 0$, the readout shift and $[B_c]_+$ are both zero.

The bound concerns readout change, not necessarily utility loss. CKA is therefore a supporting diagnostic; neither the MU profile nor its depth localization follows from CKA alone.

Side-effects for unexpected concept intervention. We separately examine changes in model behavior on high-level concepts, including Corrigible and Hallucination, from "Advanced AI Risk" [51]. In Figure 3, middle-layer patches produce larger changes in the measured concept inclinations. This is an empirical observation rather than a corollary of Proposition 1.

Potential trade-off space regarding layer stacking. Since we have one general observation that the middle layer is most fragile under unlearning, we can have two heuristic strategies to restore the specific layers from original model to maintain the general capability of LLM, which are denoted as "Before" and "After". The former utilizes unlearned layers before specific index and restores the rest layers from original model, while the latter utilizes those after specific index reversely. The results in Figure 3 show there is a potential search space for a better trade-off even for NPO with retention data.

3.3 Component-wise Replacement Unlearning

Based on our previous insights, we introduce the new Component-wise Replacement Unlearning (CRU) to partition and replace critical parts of LLM for unlearning to restore the general retention knowledge (refer to Algorithm 1). We first present a general definition and then shift our focus to the layer-wise case. For an integer $n > 0$, we let $[n] := \{1, 2, \ldots, n\}$ and we have the following.

Definition 2 (Component-wise partitioner)**.** *Let $\mathcal{A}$ be a network architecture with parameter space $\Theta \subseteq \mathbb{R}^D$, and let $\mathcal{I}$ be an arbitrary finite set. A* component-wise partitioner *is a function $\rho\colon \mathcal{I} \to [D]$ such that $\rho(I) \cap \rho(I') = \varnothing$ for any $I, I' \in \mathcal{I}$ such that $I \neq I'$. We call $\mathcal{I}$ the* index set *of ρ and $|\mathcal{I}|$ the* size *of ρ. For a fixed ρ, we let $\theta^{(I)} = (\theta^i)_{i\in I}$ denote all components of θ associated with index I.*

Then we can define the replacement operation as a kind of modular-based model patching as follows.

Definition 3 (Patched model)**.** *Given two parameters $\theta_{\mathrm{orig}}, \theta_{\mathrm{new}} \in \Theta$ and a* patching vector *$\boldsymbol{\alpha} \in \{0, 1\}^{\mathcal{I}}$, we define the* patched parameter *$\theta_{\boldsymbol{\alpha}}$ in the following component-wise manner:*

$$(\theta_{\boldsymbol{\alpha}})^I = (\theta_{\mathrm{orig}})^I, \ \text{ If } \alpha_I = 0; \text{ oth., } \ (\theta_{\boldsymbol{\alpha}})^I = (\theta_{\mathrm{new}})^I. \tag{4}$$

i.e., $\alpha_I = 0$ denotes that $\theta_{\boldsymbol{\alpha}}$ takes the same values as θ_{orig} at component I, whereas $\alpha_I = 1$ denotes that $\theta_{\boldsymbol{\alpha}}$ takes the same values as θ_{new} at component I.

In the layer-wise case, let $\mathcal{A}$ be a transformer-based architecture of LLM with parameter θ_{orig} and L layers, the *layer-wise partitioner* ρ_{layer} has an index set $\mathcal{I}_{\mathrm{layer}} = [L]$, and for any $l \in \mathcal{I}_{\mathrm{layer}}$, $\theta^{(l)}$

Table 1: Full unlearning results on TOFU across model families and sizes.

NPO	ES-exact		ES-perturb		MU↑	FQ↑	GA	ES-exact		ES-perturb		MU↑	FQ↑
	retain↑	unlearn↓	retain↑	unlearn↓				retain↑	unlearn↓	retain↑	unlearn↓		
llama3.2-1B													
Original	0.7642	0.7592	0.3286	0.3574	0.5914	-9.0517	Original	0.7642	0.7592	0.3286	0.3574	0.5914	-9.0517
Unlearned	0.0339	0.0287	0.0270	0.0281	0.2203	-2.3448	Unlearned	0.0332	0.0282	0.0265	0.0281	0.0000	-104.7672
+RT (w. $\mathcal{D}_r$)	0.1638	0.0730	0.1142	0.0700	0.4386	-2.2030	+1×KL (w. $\mathcal{D}_r$)	0.0386	0.0282	0.0303	0.0281	0.1156	-104.7672
FLAT	0.1272	0.1010	0.0993	0.0835	0.2787	-3.9575	+10×KL (w. $\mathcal{D}_r$)	0.3945	0.1214	0.1652	0.1025	0.5467	-4.3228
TNPO	0.0803	0.0373	0.0654	0.0376	0.3121	-2.2030	+20×KL (w. $\mathcal{D}_r$)	0.7360	0.3089	0.3067	0.2296	**0.5901**	-8.0218
WTNPO	0.0342	0.0287	0.0265	0.0287	0.3512	**-0.6871**	WGA	0.0340	0.0282	0.0265	0.0281	0.2898	**-0.9796**
AltPO	0.0600	0.0362	0.0599	0.0375	0.4519	-2.0646	SatImp	0.2253	0.1555	0.1457	0.1327	0.4652	-7.2902
Ours	0.2938	0.0981	0.1972	0.0851	**0.5504**	-2.0646	Ours	0.2318	0.0689	0.1362	0.0554	0.5426	-2.7916
llama3.2-3B													
Original	0.9013	0.9291	0.4241	0.4111	0.6579	-5.7157	Original	0.9013	0.9291	0.4241	0.4111	0.6579	-5.7157
Unlearned	0.0336	0.0287	0.0271	0.0281	0.0347	-7.0539	Unlearned	0.0332	0.0282	0.0265	0.0281	0.0000	-104.7672
+RT (w. $\mathcal{D}_r$)	0.1706	0.0650	0.1134	0.0678	0.4429	-1.6705	+1×KL (w. $\mathcal{D}_r$)	0.0921	0.0282	0.0663	0.0281	0.3251	-104.7672
FLAT	0.2489	0.1881	0.1481	0.1679	0.5000	-2.3448	+10×KL (w. $\mathcal{D}_r$)	0.3521	0.0575	0.1437	0.0417	0.6222	-4.7025
TNPO	0.0421	0.0282	0.0286	0.0281	0.4397	-1.4255	+20×KL (w. $\mathcal{D}_r$)	0.8340	0.4356	0.3622	0.2506	0.6633	-4.3228
WTNPO	0.0347	0.0282	0.0304	0.0281	0.4257	**-1.3084**	WGA	0.0342	0.0282	0.0277	0.0281	0.3511	**-1.3084**
AltPO	0.0356	0.0287	0.0280	0.0287	0.4899	-1.4255	SatImp	0.0341	0.0282	0.0280	0.0287	0.3120	**-1.3084**
Ours	0.0999	0.0719	0.1058	0.0846	**0.5117**	-1.5462	Ours	0.7251	0.2117	0.3677	0.1215	**0.6691**	-3.2700
llama2-7B													
Original	0.9867	0.9774	0.6018	0.5366	0.6192	-10.1446	Original	0.9867	0.9774	0.6018	0.5366	0.6192	-10.1446
Unlearned	0.0285	0.0243	0.0233	0.0238	0.0479	**-0.4366**	Unlearned	0.0278	0.0235	0.0220	0.0235	0.0000	-104.7672
+RT (w. $\mathcal{D}_r$)	0.0914	0.0267	0.1403	0.0280	0.5132	-2.3448	+1×KL (w. $\mathcal{D}_r$)	0.0512	0.0235	0.0734	0.0235	0.4980	-104.7672
FLAT	0.0278	0.0235	0.0220	0.0235	0.0000	-20.5133	+10×KL (w. $\mathcal{D}_r$)	0.4730	0.0235	0.1752	0.0235	**0.6042**	-23.9958
TNPO	0.0598	0.0313	0.0833	0.0322	0.4315	-2.6391	+20×KL (w. $\mathcal{D}_r$)	0.8473	0.3380	0.4320	0.2256	0.5934	-6.3679
WTNPO	0.0521	0.0324	0.0711	0.0336	0.4502	-2.7916	WGA	0.0405	0.0327	0.0501	0.0302	0.4037	-5.5057
AltPO	0.0604	.0330	0.0864	0.0344	0.3911	-2.0646	SatImp	0.1308	0.1295	0.2048	0.0752	0.5237	-10.1446
Ours	0.0355	0.0719	0.0309	0.0252	**0.5296**	-1.9297	Ours	0.4924	0.1131	0.2801	0.0687	0.6019	**-5.2994**
llama2-13B													
Original	1.0000	0.5727	0.9953	0.6185	0.6253	-9.3189	Original	1.0000	0.5727	0.9953	0.6185	0.6253	-9.3189
Unlearned	0.0283	0.0235	0.0233	0.0235	0.0333	-3.1070	Unlearned	0.0278	0.0235	0.0220	0.0235	0.0000	-104.7672
+RT	0.1950	0.0914	0.1624	0.0851	0.3690	**-2.2030**	+1*KL	0.0611	0.0235	0.0362	0.0235	0.3492	-104.7672
FLAT	0.8310	0.8064	0.5338	0.4917	**0.5853**	-9.8654	+10*KL	0.5991	0.0235	0.3843	0.0235	0.5747	-90.7512
TNPO	0.0949	0.0315	0.0257	0.0322	0.4468	-3.9575	+20*KL	0.7692	0.2274	0.4364	0.1745	**0.6292**	-9.8654
WTNPO	0.0291	0.0240	0.0236	0.0254	0.2319	-18.8935	WGA	0.0289	0.0235	0.0223	0.0235	0.0975	-11.9053
AltPO	0.0406	0.0310	0.0409	0.0295	0.4835	-2.4902	SatImp	0.5970	0.0235	0.3370	0.0235	0.6000	-90.7512
Ours	0.1775	0.0419	0.1368	0.0442	0.5522	-3.2700	Ours	0.4389	0.1840	0.3342	0.1507	0.5860	**-2.2030**
Phi-3.5-mini													
Original	0.9148	0.9598	0.4593	0.4078	0.6648	-7.2902	Original	0.9148	0.9598	0.4593	0.4078	0.6648	-7.2902
Unlearned	0.0272	0.0233	0.0215	0.0233	0.2874	-3.4365	Unlearned	0.0272	0.0233	0.0215	0.0233	0.0	-104.7672
+RT (w. $\mathcal{D}_r$)	0.0272	0.0233	0.0215	0.0233	0.4747	-2.0646	+1×KL (w. $\mathcal{D}_r$)	0.0273	0.0233	0.0215	0.0233	0.0016	-81.6946
FLAT	0.5361	0.4282	0.2847	0.3118	**0.6037**	-5.0968	+10×KL (w. $\mathcal{D}_r$)	0.6736	0.2525	0.2901	0.2179	0.6509	-9.8655
TNPO	0.0272	0.0233	0.0215	0.0233	0.4927	-2.6391	+20×KL (w. $\mathcal{D}_r$)	0.8907	0.5444	0.4196	0.3574	**0.6648**	-8.2735
WTNPO	0.0272	0.0233	0.0215	0.0233	0.3140	-9.0517	WGA	0.0272	0.0233	0.0215	0.0233	0.2323	-10.7151
AltPO	0.0272	0.0233	0.0215	0.0233	0.4116	-4.5108	SatImp	0.1555	0.1383	0.1077	0.1362	0.5454	**-3.1070**
Ours	0.0272	0.0233	0.0215	0.0233	0.4977	**-0.9796**	Ours	0.3117	0.1959	0.1335	0.1636	0.6245	-4.8978

denotes the parameters of the l-th layer. We have an unlearned LLM with parameter θ_{new} with vector $\boldsymbol{\alpha}$ to obtain a hybrid model $\theta_{\boldsymbol{\alpha}}$. For example, $L = 5$ and $\boldsymbol{\alpha} = [1, 0, 0, 0, 1]$ denote restoring the middle three layers of parameter from the original model to the unlearned model. The problem then can be formulated as finding a $\boldsymbol{\alpha}$ to achieve a highest score, e.g., FQ and MU for optimizing unlearning trade-off. In particular, by limiting using k layers from the unlearned model, we show a surprisingly simple solution through the newly defined score and take the top-k layer index as final $\boldsymbol{\alpha}$.

Definition 4 (Patching Score via Sorted Indices)**.** *Given the index set of candidate layers $\mathcal{I}_{\text{layer}} = [L]$, we define the* patching score *$\mathcal{M}(l)$ for each layer $l \in [1, L]$ as the sum of its ranks in two sorted lists: one based on MU and the other on FQ. Let $\mathcal{T}_{MU}(l)/\mathcal{T}_{FQ}(l)$ denote the rank index of layer l when all layers are sorted in descending order of $S_{MU}(l)/S_{FQ}(l)$ as Eq. 1. Then, the score is defined as:*

$$\mathcal{M}(l) = \mathcal{T}_{MU}(l) + \mathcal{T}_{FQ}(l). \tag{5}$$

A lower $\mathcal{M}(l)$ indicates that the layer ranks highly in both model utility and forget quality, and is thus more favorable for top-k selection in layer-wise model merging.

This top-k solution is based on the defined score without continuous assumption like previous "Before" and "After", and is verified to be effective in our scenario without exhaustive searching on the full space of size $\binom{L}{k}$, which can be further related to prior work on interchange interventions [18] and Shapley interaction [66]. Similarly, the component can be straightforwardly extended to other fine-grained ones like MLP or attention heads, **the choice on layer-wise** is on the algorithm complexity, which will be significantly increased given the enlarged search space. We provide the pseudo-code implementation with extended discussion in Appendix D. In the right of Figure 3, we also investigate **replacement v.s. structural finetuning**, i.e., whether the layer fragility can serve as an algorithmic prior to be representation regularization during finetuning, such as freezing their parameter updates for unlearning finetune. However, the results show it induces severe model performance loss, as this disrupts learning dynamics, especially when the frozen layers are critical for routing representations.

Table 2: **In MUSE, CRU achieves a better removal–retention trade-off.** Results are obtained using Llama2-7b-chat model.

Dataset	Method	ES↓	KnowMem↓ ($\mathcal{D}_t$)	VerbMem↓ ($\mathcal{D}_t$)	PrivLeak→ 0	KnowMem↑ ($\mathcal{D}_r$)
News	Original	0.3503	0.4471	0.6399	-96.86	0.4470
	Unlearned (NPO)	0.0222	0.3433	0.1500	-63.86	0.3090
	+RT (w. $\mathcal{D}_r$, NPO)	0.0669	0.3816	0.2653	-93.19	0.4458
	Ours	0.0289	0.3673	0.1609	-76.43	0.4443
	Original	0.3503	0.4471	0.6399	-96.86	0.4470
	Unlearned (GA)	0.0079	0.0000	0.0000	56.61	0.0000
	+KL (w. $\mathcal{D}_r$, GA)	0.0083	0.3607	0.0589	80.18	0.1893
	Ours	0.0225	0.1656	0.1483	-67.53	0.3294
Books	Original	0.9228	0.4878	0.9962	-56.93	0.7113
	Unlearned (NPO)	0.8274	0.4298	0.9550	-59.24	0.5361
	+RT (w. $\mathcal{D}_r$, NPO)	0.8667	0.4067	0.9175	-56.00	0.7078
	Ours	0.8397	0.3777	0.9351	-57.85	0.5540
	Original	0.9228	0.4878	0.9962	-56.93	0.7113
	Unlearned (GA)	0.0079	0.0000	0.0000	5.59	0.0000
	+KL (w. $\mathcal{D}_r$, GA)	0.1079	0.1235	0.2731	-64.80	0.1613
	Ours	0.0526	0.1505	0.2047	-61.56	0.6108

Table 3: **Comparison on WMDP** for CRU with Different LLMs.

Method	WMDP↓ (similar to FQ)	MMLU↑ (similar to MU)
Llama3.2-1B		
Original	0.3533	0.4694
GA	0.2431	0.2465
NPO	0.2582	0.2329
Ours	0.2864	0.3902
Llama3.2-3B		
Original	0.4046	0.6221
GA	0.2431	0.2465
NPO	0.2587	0.2291
Ours	0.2652	0.3735
Zephyr-7b		
Original	0.4288	0.5880
GA	0.2455	0.2689
NPO	0.2456	0.2551
Ours	0.2647	0.4900

Table 4: **On RESTOR, CRU improves accuracy** over GA, KL, and NPO across corruption levels [55].

Clean		Corrupt	GA	GA + CRU	KL	KL + CRU	NPO	NPO + CRU
72.20	k = 1	68.56	51.09	**78.17**	65.94	**78.60**	76.00	**79.04**
	k = 2	67.69	55.90	**74.24**	69.43	**75.55**	76.86	**79.04**
	k = 3	65.50	48.91	**73.36**	69.00	**73.36**	74.24	**77.29**
	k = 4	59.83	51.09	**65.94**	66.38	**73.36**	74.55	**76.89**
	k = 5	57.64	57.20	**66.81**	66.38	**71.62**	74.24	**76.86**

Table 5: **Forgetting and retention** on TOFU (ReLearn-style).

Method	KFR↑	KRR↑
Original	0.350	0.635
NPO	0.950	0.095
+ CRU ($k = 4$)	0.925	0.300
+ CRU ($k = 9$)	0.900	0.175

4 Experiment

4.1 Experimental Setups

Datasets. In our experiments, we mainly explore unlearning methods using the Task of Fictitious Unlearning (TOFU) dataset [34], which serves as a representative benchmark in previous works [91, 80, 79]. To evaluate generalization across distinct unlearning regimes, we additionally adopt three benchmarks: MUSE [60] and WMDP [28] cover removal of news, book information, and hazardous chemical knowledge; RESTOR [55] probes the data-level restorative ability of unlearning by measuring how well an unlearned model recovers the behavior of a counterfactual "clean" model after a set of k corrupted facts. More details are in Appendix E.1.

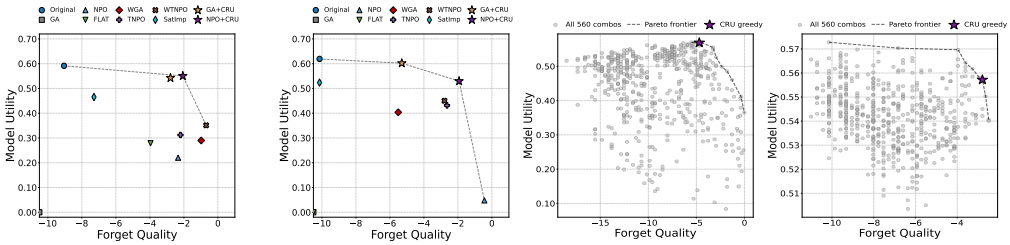
Evaluation protocol. On TOFU, we use the Forget05 setting, with Forget01 and Holdout01 reserved for testing. The remaining Forget04 and Holdout04 form the validation set; validation and test authors are disjoint. On MUSE, we shuffle each evaluation subset and split it into 80% validation and 20% test data. For these benchmarks, CRU selects layers using validation performance, and we report results on the test splits. Full metric definitions are provided in Appendix E.1.

Unlearning baselines. To verify the effectiveness of our methods in general scenarios, we consider 2 representative baselines, e.g., GA [85], NPO [91], 4 recent advanced methods based on them, e.g., Weighted Gradient Ascent (WGA), Token-wise NPO (TNPO), Weighted Token-wise NPO (WTNPO) [79], Forget data only Loss AdjustmenT (FLAT) [80], +KL/+RT with retention data on GA/NPO, and 2 regularization-based methods, AltPO [39] and SatImp [82], for comparison with the same setups. We leave full description of the baselines in Appendix E.2.

Implementation details. For TOFU, we use Llama3.2-1B-Instruct model, Llama3.2-3B-Instruct model [20], Llama2-7b-chat model, Llama2-13b-chat model [71] and Phi-3.5-mini model [1]. For MUSE, we adopt Llama2-7b-chat model. For WMDP, we additionally adopt Zephyr-7b model[72]. For most TOFU and MUSE experiments, we use AdamW with a learning rate of 10^{-5}, an effective batch size of 32, and 10 unlearning epochs. Method-specific hyperparameters and exceptions are detailed in Appendix E.3.

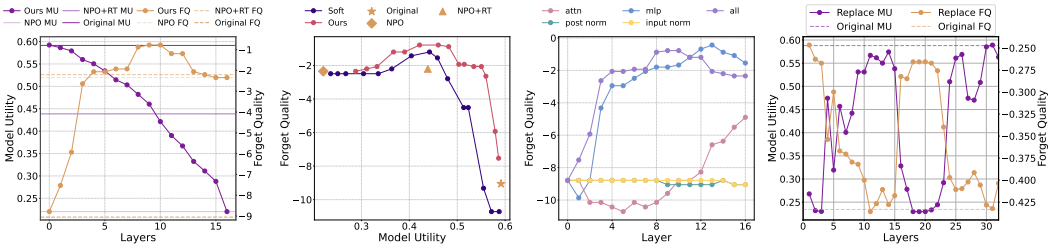
4.2 Main Comparison

In Tables 1, 2 and 3, we summarize the performance on TOFU, MUSE, and WMDP. The overall results include CRU compared with a series advanced designs based on NPO [91] (on the left side: +RT, FLAT, TNPO, WTNPO, AltPO) and GA [85] (on the right side:, +KL of different strength, WGA, SatImp) with the original models (pre-unlearned). To facilitate reading, we only mark the best results under primary metrics such as MU and FQ in Table 1, where the other ES-related metrics are



(a) 1B trade-off (b) 7B trade-off (c) GA (k=3) (d) NPO (k=3)

Figure 4: **CRU sits on the FQ–MU Pareto frontier and matches the exhaustive optimum. (a, b)** GA+CRU and NPO+CRU dominate baselines on the Pareto frontier. **(c, d)** CRU greedy (purple star) lands at the upper-right Pareto corner of all 560 three-layer combinations on Llama3.2-1B.



(a) Selected layers (b) Hard vs. soft (c) Components (d) WMDP

Figure 5: **Ablation studies and fragility patterns.** **(a)** Performance with different numbers of selected layers k. **(b)** Hard replacement versus soft intervention. **(c)** Fragility across fine-grained transformer components. **(d)** Layer-wise fragility on WMDP.

fine-grained results. We also mark the methods including retention data with (w. $\mathcal{D}_r$). We additionally compare CRU with RMU on TOFU in Table 11.

Can CRU achieve better a performance trade-off? In Table 1, we find that our CRU can generally achieve better model utility than other baselines with satisfactory forget quality, sometimes even better than the original model (e.g., in llama3.2-3B based on GA). Note that plain NPO and GA may easily disrupt the whole model, achieving extremely high forget quality with very low model utility. Without directly changing the training process, our post-hoc component replacement can still restore the natural functionality of LLM after unlearning, it is also validated in later qualitative results. We further visualize this trade-off in Figure 4(a, b): on the FQ–MU plane for Llama3.2-1B and Llama2-7B, GA+CRU and NPO+CRU sit on the Pareto frontier above all unlearning baselines, confirming that the gains are substantial rather than marginal. In addition, CRU exhibits favorable computational efficiency compared with the baselines (see Figure 12) produces more coherent generations than GA/NPO (Table 12), and changes fewer middle-layer parameters than other baselines (Figure 14).

Whether simply including retention data can be a better solution? In Table 1, we also consider the comparison with including retention data during unlearning, e.g., adding the NLL loss in retention data (+RT) with NPO or adding the KL loss of the original model output (+KL) with GA, which is a straightforward solution to control the excessive unlearning. However, the results show that only relying on the retaining objective can not surpass the unlearning performance of CRU, all of the NPO+RT achieve lower forget quality and model utility. On the GA-side, we also enhance the strength of KL regularization, although it can indeed boost the model utility to reach a similar state with CRU, the FQ is also significantly affected, indicating it is non-trivial to optimize the trade-off.

How the method varied across different models and unlearning benchmarks? Beyond TOFU, we evaluate CRU on MUSE, WMDP, and RESTOR. On MUSE (Table 2), CRU better suppresses forgetting content (KnowMem on $\mathcal{D}_t$) while preserving retention (KnowMem on $\mathcal{D}_r$), outperforming base methods even with retaining objectives added. On WMDP (Table 3), CRU achieves lower WMDP with significantly higher MMLU. On the data-level restorative regime (Table 4), applying CRU on top of GA, KL, and NPO consistently improves accuracy across corruption levels $k \in \{1, \ldots, 5\}$,

indicating that CRU's layer-wise restoration generalizes from the FQ–MU trade-off to recovering counterfactual "clean" behavior. Appendix E.6 compares additional WMDP training setups.

4.3 Further Exploration and Ablation Studies

Rank-sum greedy selection is near-optimal. On Llama3.2-1B (16 layers), we exhaustively enumerate all $\binom{16}{3}$=560 three-layer replacement combinations under GA and NPO. In Figure 4(c, d), CRU's greedy selection (purple star) lies at the upper-right Pareto corner of this search. This supports near-optimal selection using L single-layer evaluations rather than evaluating $\binom{L}{k}$ combinations.

Selected layer numbers. In Figure 5(a), we plot the number of layers k considered in our selection versus performance change. When $k = 0$, the final model is same as the original one; when the $k = 16$ (the total number of LLM layers), the final model approaches the unlearned model. CRU can obtain a hybrid model with better tradeoff (even surpass that of NPO+RT) with k from 4-9 in Llama3.2-1B-Instruct, which is also suitable for other larger models. In practice, users can specify utility and forgetting thresholds according to their unlearning requirements and select k using validation data, with all criteria fixed before test evaluation. We provide example selection strategies and validation-stability diagnostics in Appendices D and E.4.

Compare replacement to soft intervention. Instead of selecting the replaced layers based on validation, we also consider another coarse operation to obtain the final model, i.e., a soft intervention with the original model as $\theta^{\mathrm{r}} = \epsilon \cdot \theta + (1 - \epsilon) \cdot \theta^{\mathrm{u}}$ in the model level. In Figure 5(b), we compare the performance trade-off between CRU, soft interventions, and also other baseline indicators. The results demonstrate that our method can surpass the soft intervention with a whole model using different $\epsilon \in [0, 1]$, and also validate the advantages of our layer-wise selection.

Fragility across transformer components. Figure 5(c) compares component-wise replacement on Llama3.2-1B unlearned using NPO. MLP replacement follows a similar FQ trend to full-layer replacement, whereas input/post-normalization has limited effects and attention replacement exhibits a different trend. The corresponding MU results show a similar distinction between MLP/full-layer replacement and the other components (Figure 8). We conjecture that these differences reflect the components' functional roles: MLPs transform and re-encode intermediate representations, making their updates more influential for the removal–retention trade-off; normalization primarily stabilizes representations; and attention routes information across tokens, with effects distributed across layers and heads. This interpretation motivates treating the full transformer layer as a practical unit for diagnosing fragility, while recognizing that its internal components contribute differently.

Fragility across benchmarks. Figure 5(d) extends layer-wise patching to the GA-unlearned Zephyr-7B model on WMDP. A broadly similar U-shaped trend is observed, although the curves contain sharp excursions and the locations of fragile layers differ from TOFU. Together with the TOFU results across model sizes and unlearning objectives, this supports a recurring, non-uniform depth profile. We interpret middle-layer fragility in a relative, benchmark-dependent sense, rather than as a fixed layer index or an identical curve across settings. These variations motivate validation-based layer selection rather than imposing a fixed replacement interval.

Knowledge recovery risk. Table 5 complements MUSE: CRU improves retention while maintaining high, slightly reduced forgetting rates. Soft-token extraction and frozen white-box probes show increased target accessibility over NPO, below Original; targeted relearning reveals greater early recovery under limited re-exposure (Appendix E.5). Thus, utility restoration can also recover target information. Together with the varying-k and soft-intervention results, these audits motivate balancing retention and forgetting during validation. CRU improves this trade-off empirically, but neither utility recovery nor high behavioral forgetting rates certify deletion or resistance to future attacks.

5 Conclusion

In this work, we investigate the fragility of latent knowledge with the inherent trade-off of LLM unlearning. Introducing a unified analytical approach based on layer-wise patching, we isolate and characterize the effects on LLM internal representation under unlearning, and reveal the non-uniform influence from different layers on the validation performance degradation. Such effects align with different levels of abstraction encoded in LLMs. Based on these insights, we propose a lightweight and general framework called CRU, opening the new possibilities for surgical unlearning.

Acknowledgments and Disclosure of Funding

The authors would like to express their sincere gratitude to the anonymous reviewers and the area chairs for their thorough review and constructive feedback. Their insightful comments and valuable suggestions have significantly enhanced the quality and clarity of this manuscript. We deeply appreciate their time and effort in helping us improve our work.

References

[1] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

[4] Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. Steering large language model activations in sparse spaces. *arXiv preprint arXiv:2503.00177*, 2025.

[5] Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*, 2024.

[6] Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice). *Advances in Neural Information Processing Systems*, 37:84298–84328, 2024.

[7] Angie Boggust, Hyemin Bang, Hendrik Strobelt, and Arvind Satyanarayan. Abstraction alignment: Comparing model-learned and human-encoded conceptual relationships. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2025.

[8] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *S&P*, 2021.

[9] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *ICLR*, 2023.

[10] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *USENIX Security*, 2023.

[11] Ting-Yun Chang, Jesse Thomason, and Robin Jia. Do localization methods actually localize memorized data in llms? a tale of two benchmarks. In *NAACL-HLT*, 2024.

[12] DeepSeek-AI. Deepseek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[14] Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zachary C Lipton, J Zico Kolter, and Pratyush Maini. OpenUnlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics. *arXiv preprint arXiv:2506.12618*, 2025.

[15] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

[16] Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *arXiv preprint arXiv:2410.07163*, 2024.

[17] Yujie Feng, Xu Chu, Yongxin Xu, Guangyuan Shi, Bo Liu, and Xiao-Ming Wu. TaSL: Continual dialog state tracking via task skill localization and consolidation. In *ACL*, pages 1266–1279, 2024.

[18] Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning*, pages 7324–7338. PMLR, 2022.

[19] 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.

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

[21] Kaarel Hänni, Jake Mendel, Dmitry Vaintrob, and Lawrence Chan. Mathematical models of computation in superposition. *arXiv preprint arXiv:2408.05451*, 2024.

[22] Yifei He, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao. Localize-and-stitch: Efficient model merging via sparse task arithmetic. *arXiv preprint arXiv:2408.13656*, 2024.

[23] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

[24] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[25] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 1998.

[27] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530, 2023.

[28] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *ICML*, 2024.

[29] Yuxiao Li, Eric J Michaud, David D Baek, Joshua Engels, Xiaoqing Sun, and Max Tegmark. The geometry of concepts: Sparse autoencoder feature structure. *Entropy*, 27(4):344, 2025.

[30] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *ACL*, 2004.

[31] 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*, pages 1–14, 2025.

[32] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, et al. AgentBench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*, 2023.

[33] Tinh Son Luong, Thanh-Thien Le, Linh Ngo Van, and Thien Huu Nguyen. Realistic evaluation of toxicity in large language models. *arXiv preprint arXiv:2405.10659*, 2024.

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

[35] Pratyush Maini, Michael Curtis Mozer, Hanie Sedghi, Zachary Chase Lipton, J Zico Kolter, and Chiyuan Zhang. Can neural network memorization be localized? In *International Conference on Machine Learning*, pages 23536–23557. PMLR, 2023.

[36] Valentino Maiorca, Luca Moschella, Antonio Norelli, Marco Fumero, Francesco Locatello, and Emanuele Rodolà. Latent space translation via semantic alignment. *Advances in Neural Information Processing Systems*, 36:55394–55414, 2023.

[37] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[38] Riccardo Massidda, Atticus Geiger, Thomas Icard, and Davide Bacciu. Causal abstraction with soft interventions. In *Conference on Causal Learning and Reasoning*, pages 68–87. PMLR, 2023.

[39] Anmol Mekala, Vineeth Dorna, Shreya Dubey, Abhishek Lalwani, David Koleczek, Mukund Rungta, Sadid Hasan, and Elita Lobo. Alternate preference optimization for unlearning factual knowledge in large language models. *arXiv preprint arXiv:2409.13474*, 2024.

[40] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *NeurIPS*, volume 35, pages 17359–17372, 2022.

[41] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.

[42] Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. *arXiv preprint arXiv:2311.12983*, 2023.

[43] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. In *ICLR*, 2023.

[44] Trung Nguyen and Yan Leng. Toward a flexible framework for linear representation hypothesis using maximum likelihood estimation. *arXiv preprint arXiv:2502.16385*, 2025.

[45] Kerem Oktar, Ilia Sucholutsky, Tania Lombrozo, and Thomas L Griffiths. Dimensions of disagreement: Unpacking divergence and misalignment in cognitive science and artificial intelligence. *arXiv preprint arXiv:2310.12994*, 2023.

[46] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[47] Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

[48] Jayneel Parekh, Pegah Khayatan, Mustafa Shukor, Alasdair Newson, and Matthieu Cord. A concept-based explainability framework for large multimodal models. *Advances in Neural Information Processing Systems*, 37:135783–135818, 2024.

[49] Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. *arXiv preprint arXiv:2406.01506*, 2024.

[50] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

[51] Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13387–13434, 2023.

[52] 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.

[53] Goutham Rajendran, Simon Buchholz, Bryon Aragam, Bernhard Schölkopf, and Pradeep Kumar Ravikumar. From causal to concept-based representation learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[54] Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. In *2023 ieee conference on secure and trustworthy machine learning (satml)*, pages 464–483. IEEE, 2023.

[55] Keivan Rezaei, Khyathi Chandu, Soheil Feizi, Yejin Choi, Faeze Brahman, and Abhilasha Ravichander. Restor: Knowledge recovery in machine unlearning. *Transactions on Machine Learning Research*, 2025.

[56] Jeffrey Rosen. The right to be forgotten. *Stan. L. Rev. Online*, 64:88, 2011.

[57] Mansi Sakarvadia, Aswathy Ajith, Arham Mushtaq Khan, Nathaniel C Hudson, Caleb Geniesse, Kyle Chard, Yaoqing Yang, Ian Foster, and Michael W Mahoney. Mitigating memorization in language models. In *ICLR*, 2025.

[58] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.

[59] William F Shen, Xinchi Qiu, Meghdad Kurmanji, Alex Iacob, Lorenzo Sani, Yihong Chen, Nicola Cancedda, and Nicholas D Lane. Lunar: Llm unlearning via neural activation redirection. *arXiv preprint arXiv:2502.07218*, 2025.

[60] 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*, 2024.

[61] Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Gergely Szilvasy, Rich James, Xi Victoria Lin, Noah A. Smith, Luke Zettlemoyer, Scott Yih, and Mike Lewis. In-context pretraining: Language modeling beyond document boundaries. In *ICLR*, 2024.

[62] Oscar Skean, Md Rifat Arefin, Yann LeCun, and Ravid Shwartz-Ziv. Does representation matter? exploring intermediate layers in large language models. *arXiv preprint arXiv:2412.09563*, 2024.

[63] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

[64] Ilia Sucholutsky and Tom Griffiths. Alignment with human representations supports robust few-shot learning. *Advances in Neural Information Processing Systems*, 36:73464–73479, 2023.

[65] Ilia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim, Bradley C Love, Erin Grant, Iris Groen, Jascha Achterberg, et al. Getting aligned on representational alignment. *arXiv preprint arXiv:2310.13018*, 2023.

[66] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pages 9269–9278. PMLR, 2020.

[67] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[68] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 303–319. IEEE, 2022.

[69] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4007–4022, 2022.

[70] Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*, 2023.

[71] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[72] Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. In *First Conference on Language Modeling*, 2024.

[73] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv e-prints*, pages arXiv–2308, 2023.

[74] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

[75] Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio Ansuini, and Alberto Cazzaniga. The geometry of hidden representations of large transformer models. *Advances in Neural Information Processing Systems*, 36:51234–51252, 2023.

[76] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[77] Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. In *International Conference on Machine Learning*, pages 50268–50287. PMLR, 2024.

[78] Qizhou Wang, Bo Han, Puning Yang, Jianing Zhu, Tongliang Liu, and Masashi Sugiyama. Towards effective evaluations and comparisons for llm unlearning methods. In *The Thirteenth International Conference on Learning Representations*, 2025.

[79] Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q. Weinberger. Rethinking llm unlearning objectives: A gradient perspective and go beyond. In *ICLR*, 2025.

[80] Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Parag Shah, Yujia Bao, Yang Liu, and Wei Wei. Llm unlearning via loss adjustment with only forget data. In *ICLR*, 2025.

[81] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *NeurIPS*, 2023.

[82] Puning Yang, Qizhou Wang, Zhuo Huang, Tongliang Liu, Chengqi Zhang, and Bo Han. Exploring criteria of loss reweighting to enhance llm unlearning. *arXiv preprint arXiv:2505.11953*, 2025.

[83] Puning Yang, Qizhou Wang, Zhuo Huang, Tongliang Liu, Chengqi Zhang, and Bo Han. Exploring criteria of loss reweighting to enhance llm unlearning. *arXiv preprint arXiv:2505.11953*, 2025.

[84] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Eric Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *arXiv preprint arXiv:2312.02003*, 2023.

[85] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *arXiv preprint arXiv:2310.10683*, 2023.

[86] Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*, 2023.

[87] Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. Knowledge circuits in pretrained transformers. *arXiv preprint arXiv:2405.17969*, 2024.

[88] Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning? *arXiv preprint arXiv:2502.14010*, 2025.

[89] Dawen Zhang, Pamela Finckenberg-Broman, Thong Hoang, Shidong Pan, Zhenchang Xing, Mark Staples, and Xiwei Xu. Right to be forgotten in the era of large language models: Implications, challenges, and solutions. *arXiv preprint arXiv:2307.03941*, 2023.

[90] Jinyang Zhang, Yue Fang, Hongxin Ding, Weibin Liao, Muyang Ye, Junfeng Zhao, Yasha Wang, and Xu Chu. ADEPT: Continual pretraining via adaptive expansion and dynamic decoupled tuning. In *ICLR*, 2026.

[91] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *COLM*, 2024.

[92] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

[93] Zhanke Zhou, Chentao Cao, Xiao Feng, Xuan Li, Zongze Li, Xiangyu Lu, Jiangchao Yao, Weikai Huang, Tian Cheng, Jianghangfan Zhang, Tangyu Jiang, Linrui Xu, Yiming Zheng, Brando Miranda, Tongliang Liu, Sanmi Koyejo, Masashi Sugiyama, and Bo Han. Alphaapollo: A system for deep agentic reasoning. *arXiv preprint arXiv:2510.06261*, 2025.

[94] Zhanke Zhou, Zongze Li, Weikai Huang, Xuan Li, Chentao Cao, Xiao Feng, Xiangyu Lu, Jinbo Hu, Yi Xie, Nico Pelleriti, Shiyang Liu, Menghan Lu, Max Zimmer, Brando Miranda, Jiangchao Yao, Bo Liu, Sanmi Koyejo, Sebastian Pokutta, and Bo Han. Alphadiana: A measurement system for agents in verifiable reasoning tasks. *arXiv preprint arXiv:2610.20675*, 2026.

[95] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

[96] Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

Appendix

The appendix is organized as follows. In Appendix A, we provide reproducibility details and a link to the implementation. In Appendix B, we provide a comprehensive discussion of related work. In Appendix C, we conduct formal analysis on the latent knowledge fragility with representation drift. In Appendix D, we present the detailed implementation and extension of our component-wise replacement unlearning. In Appendix E, we provide the supplementary experimental results. Appendix F discusses broader impacts and limitations.

LLM Usage Statement

Here we clarify the usage of Large Language Models (LLMs) in this work. For the preparation of this paper, LLMs (e.g., ChatGPT) are limited to the role of a general-purpose writing assistant and are not used for research ideation or core content generation. For research purposes, LLMs were our core study subject as indicated in our title and research content, and introduced in experimental sections.

A Reproducibility Statement

The implementation is available at `https://github.com/tmlr-group/CRU`. Reproducibility details follow:

- **Datasets.** The unlearning benchmarks (e.g., TOFU [34], MUSE [60] and WMDP [28]) we used are all publicly accessible, which is introduced in Section 4.1 and Appendix E.1.
- **Assumption.** Following the previous work, we set our experiments to a tuning scenario where a well-trained LLM is available that trained on target data or contain specific knowledge.
- **Open source.** The implementation is built upon OpenUnlearning [14].
- **Environment.** All experiments are conducted with multiple runs on NVIDIA-A100-80GB GPUs with Python 3.11 and PyTorch 2.4.1. More detailed requirements can also be found in the environment descriptions in our aforementioned source codes.

B Detailed Related Work

Here we discuss related work from several aspects, e.g., LLM unlearning, mechanistic interpretability, representation geometry and concept intervention, model merging and representation alignment.

LLM Unlearning. Machine unlearning seeks to remove specific information from a trained model without full retraining. In classic settings, early works focused on algorithmic formulations, efficiency, and auditability mainly in classification models. Specifically, they introduced exact unlearning in convex models [19], certified data removal [8], gradient ascent-based forgetting [69, 68, 58], and also broader surveys [8] that summarized challenges and approaches. With the rise of LLMs, recent efforts have shifted toward scalable and reliable unlearning approaches, such as Negative Preference Optimization [91, 16] (derived from Direct Preference Optimization [52]), loss adjustment [80, 79, 78], and neural activation redirection [59]. The emerging research direction is important for ensuring the safe deployment of foundation models [28, 31]. Several works also propose various benchmarks with different evaluation metrics such as TOFU [34], MUSE [60], and the unified framework of OpenUnlearning [14]. However, most existing methods operate on the level of gradients or loss terms for objective-level adjustment, lacking understanding on how target knowledge is encoded within the model. Our work departs from previous work by treating the LLM internals itself as a functional composition of modular units. By introducing the selective patching approach, we uncover a layer-wise map of knowledge fragility that benefits preserving core functionalities under unlearning.

Post-Training for Capability Enhancement and System Evaluation. Beyond privacy-motivated unlearning, post-training is also used to strengthen model capabilities. For example, DeepSeek-R1 shows that reinforcement learning can elicit stronger reasoning behaviors, including self-reflection and verification, on verifiable tasks [12]. Systems such as AlphaApollo extend this direction to long-horizon agentic reasoning by combining multi-turn interaction, tool-use learning, and iterative solution

refinement with tool-assisted verification [93]. Evaluating these gains requires looking beyond a model's standalone answers: AgentBench tests LLMs as agents across interactive environments, while GAIA evaluates general assistants on tasks requiring capabilities such as reasoning, web browsing, multimodal understanding, and tool use [32, 42]. AlphaDiana complements benchmark-level comparisons with controlled measurements of complete agent setups and interventions on tools, skills, and memory [94]. CRU addresses a different post-training objective: after unlearning, it selects components for restoration to recover utility while maintaining the desired forgetting effect.

Mechanistic Interpretability of Transformers. Transformer models exhibit distinct functionalities across layers. Probing studies and patching experiments have revealed the localization of factual knowledge in intermediate feedforward modules. Recent progress in mechanistic interpretability has advanced our understanding of how transformer models encode, process, and reuse information internally. The seminal work on induction heads [46] identifies specific attention patterns responsible for in-context learning by modeling token repetition dynamics. Building on this, [88] further disentangles the contribution of different attention heads to in-context capabilities, revealing layer- and task-specific specialization. Broader reviews such as [5, 54] systematize techniques for probing and attributing functional roles to components within deep networks, emphasizing their importance for AI safety and transparency. Extending mechanistic approaches to multimodal settings, [6] introduces sparse linear concept embeddings to interpret internal representation space of pre-trained vision language model, while [48] proposes a general concept-based explainability framework for large vision-language models. Together, these works underscore the growing interest in aligning internal model mechanisms with human-interpretable abstractions across both language and multimodal domains. Our method complements these analyses by decomposing layer-wise forgetting–utility trade-offs.

Representation Geometry and Concept Intervention. LLM representations are highly entangled and complex. Exploring the representation geometry has gained increasingly attention recently in order to understand the role of LLM internals in concept encoding and intervention. The linear representation hypothesis has emerged as a central perspective, positing that abstract concepts are embedded in approximately linear subspaces within model activations [50, 44]. Several works have explored the geometry of these representations in general latent space, revealing structured manifolds associated with syntax and hierarchy [49, 75, 29, 63]. Probing intermediate layers has shown that key information is often concentrated in specific layers and dimensions, motivating both analysis and control strategies [62]. On the intervention side, recent works such as activation addition and contrastive activation engineering [47, 74, 73, 4] demonstrate the ability to steer model outputs by modifying internal activations, particularly in sparse or localized directions. These approaches are complemented by inference-time interventions [27] and concept-based representation learning frameworks [53], which aim to manipulate model behaviors via interpretable latent directions. In contrast, our component-wise replacement unlearning focuses not on steering outputs through activation modification, but on isolating and quantifying the functional contribution of different model components. Rather than searching for explicit concept vectors or sparse directions, our method reveals implicit knowledge fragility for preserving utility under unlearning, offering a complementary approach rooted in architectural dissection rather than intervention.

Model Merging and Representation Alignment. As neural networks become increasingly modular and over-parameterized, aligning and integrating their internal representations has emerged as a crucial problem for knowledge composition and transfer. Recent research on model merging and representation alignment has explored how neural networks encode and align information across different tasks and modalities. Early foundational work revisited the similarity of neural network representations [25], introducing metrics like CKA to quantify alignment in learned features. Building on this, studies such as [64, 65, 45, 7] compare model representations with human conceptual spaces, revealing the benefits of aligning abstractions for improved generalization and interpretability. Recent works have proposed methods for latent space translation [36] and zero-shot communication [43], leveraging relative or semantic alignment to facilitate knowledge transfer across models. Furthermore, a series of efforts target model merging through structured alignment: [77] highlights the importance of identifying task-relevant subspaces for merging, while [22] proposes sparse, component-wise arithmetic to achieve efficient fusion across model variants. Our component-wise replacement unlearning differs from these approaches by focusing not on merging models to aggregate or transfer capabilities, but on isolating and suppressing specific knowledge.

Model Editing and Component-Level Adaptation. Model editing methods seek to change specific associations in a pretrained model. ROME uses causal tracing to locate computations that mediate factual recall and applies a rank-one update to edit a selected association [40]. Related work studies localized components for acquiring or retaining task capabilities: TaSL identifies task-specific and task-shared skill units and consolidates them for continual dialogue-state tracking [17], while ADEPT uses adaptive layer expansion and decoupled tuning for domain-adaptive continual pretraining [90]. These methods show how localized interventions can support targeted editing or continual adaptation. CRU addresses a complementary setting: given original and unlearned checkpoints, it diagnoses utility damage and composes selected components post hoc to recover utility while preserving forgetting, without further training. Whole-layer replacement provides a practical unit for checkpoint composition, with validation measurements guiding the selection.

Differences. Building on these perspectives, our work introduces latent knowledge fragility as an unlearning-specific lens for analyzing the propagation of functional changes across model components. Checkpoint composition is the execution mechanism of CRU; its focus is validation-guided component restoration to improve the forgetting–utility trade-off after unlearning. This connects localization and model composition to unlearning-specific diagnosis and repair.

C Formal Analysis of Latent Knowledge Fragility

We prove Proposition 1 via representation drift and readout alignment; r denotes subspace dimension, distinct from the layer budget k.

Assumption 1 (Common concept subspace and readout)**.** *Let $\Phi_\ell^{\text{orig}}, \Phi_\ell^{\text{unlearn}} \in \mathbb{R}^{n\times d}$ be centered hidden representations of the same ordered $n \geq 2$ retained examples. Fix $P_c \in \mathbb{R}^{d\times r}$, $1 \leq r \leq d$, and nonzero $Z_o = \Phi_\ell^{\text{orig}} P_c$, $Z_u = \Phi_\ell^{\text{unlearn}} P_c$. A common nonzero $W_c \in \mathbb{R}^r$ and $b \in \mathbb{R}$ define $g_c^s = Z_s W_c + b\mathbf{1}$ for $s \in \{o, u\}$, with $\mathbf{1} \in \mathbb{R}^n$ the all-ones vector. All projections, readouts, biases, and coordinates are fixed.*

Proposition 2 (CKA and functional change under readout alignment)**.** *Under Assumption 1, let $\Delta Z = Z_u - Z_o$, $\sigma_c^s = \|Z_s\|_F/\sqrt{n}$, and*

$$\text{CKA}_c = \frac{\|Z_o^\top Z_u\|_F^2}{\|Z_o^\top Z_o\|_F \|Z_u^\top Z_u\|_F}.$$

With B_c defined in Eq. 3, the representation drift satisfies $\|\Delta Z\|_F^2/n \geq [B_c]_+$. If $\Delta Z \neq 0$ and its readout alignment satisfies

$$\kappa_\ell = \frac{\|\Delta Z W_c\|_2^2}{\|W_c\|_2^2 \|\Delta Z\|_F^2} \geq \gamma > 0,$$

then $\|g_c^u - g_c^o\|_2^2/n \geq \gamma \|W_c\|_2^2 [B_c]_+$. For zero drift, both sides vanish without defining κ_ℓ.

Proof. **Representation drift.** Expanding the squared Frobenius norm gives

$$\frac{1}{n}\|\Delta Z\|_F^2 = (\sigma_c^o)^2 + (\sigma_c^u)^2 - \frac{2}{n}\operatorname{tr}(Z_o^\top Z_u).$$

For $A = Z_o^\top Z_u \in \mathbb{R}^{r\times r}$, Cauchy–Schwarz in the Frobenius inner product yields

$$|\operatorname{tr}(A)| = |\langle I_r, A\rangle_F| \leq \|I_r\|_F \|A\|_F = \sqrt{r}\|A\|_F,$$

where I_r is the identity matrix. The definition of CKA gives

$$\|A\|_F = \sqrt{\text{CKA}_c}\left(\|Z_o^\top Z_o\|_F \|Z_u^\top Z_u\|_F\right)^{1/2}.$$

Substitution and nonnegativity give $\|\Delta Z\|_F^2/n \geq [B_c]_+$.

Readout change. For $\Delta Z \neq 0$, the common bias cancels, and the alignment assumption gives

$$\frac{1}{n}\|g_c^u - g_c^o\|_2^2 = \kappa_\ell \|W_c\|_2^2 \frac{\|\Delta Z\|_F^2}{n} \geq \gamma \|W_c\|_2^2 [B_c]_+.$$

For $\Delta Z = 0$, the drift bound gives $B_c \leq 0$ and the readout shift is zero. □

Why alignment is required. Cauchy–Schwarz gives $0 \le \kappa_\ell \le 1$, but no positive lower bound. For $n \ge 3$, take nonzero centered $a, c \in \mathbb{R}^n$ with $a^\top c = 0$, $Z_o = (a, 0)$, $Z_u = (c, 0)$, and $W_c = (0, 1)^\top$. Then $\mathrm{CKA}_c = 0$ and $B_c > 0$, yet $\Delta Z W_c = 0$. Low CKA alone need not change the readout.

Approximate readouts. Suppose scalar outputs obey $y_c^s = g_c^s + e_s$, with $e_s \in \mathbb{R}^n$ and $\varepsilon = \|e_u - e_o\|_2/\sqrt{n}$. The reverse triangle inequality and Proposition 2 give

$$\frac{1}{n}\|y_c^u - y_c^o\|_2^2 \ge \left[\sqrt{\gamma \|W_c\|_2^2 [B_c]_+} - \varepsilon\right]_+^2. \tag{6}$$

Indeed, $\|y_c^u - y_c^o\|_2/\sqrt{n} \ge [\|g_c^u - g_c^o\|_2/\sqrt{n} - \varepsilon]_+$, and positive part is nondecreasing. If $\|e_s\|_2/\sqrt{n} \le \varepsilon_s$, then $\varepsilon \le \varepsilon_o + \varepsilon_u$. Actual-output bounds therefore require control of the approximation residuals.

Interpretation and scope. Holding σ_c^o, σ_c^u, covariance-norm factors, W_c, and γ fixed, lower CKA raises the readout-shift lower bound, which may be zero. This does not guarantee accuracy or MU degradation, determine Eq. 1, or imply universal middle-layer fragility. Figure 3 remains empirical evidence; applying Eq. 6 to actual outputs additionally requires controlled approximation error.

Empirical readout-alignment diagnostic. We examine one operational realization of the alignment assumption using Original and NPO checkpoints of Llama3.2-1B. The diagnostic uses 320 TOFU retain questions from 16 authors, each with one correct paraphrased answer and five perturbed answers. At each layer, we extract the mean causal pre-answer hidden state for each answer candidate. We use four author-disjoint folds: the PCA projection P_c and a correct-answer linear readout W_c are fitted only on Original training-author features, then held fixed when comparing Original and NPO on held-out authors. The subspace dimension r is distinct from the CRU layer budget k.

Table 6: Readout-alignment diagnostics on retained TOFU questions. Accuracy is Original's held-out six-answer choice accuracy, averaged over 16 layers. For each layer, $\bar{\kappa}_\ell$ pools numerator and denominator contributions from the four held-out folds; the minimum and mean below are across these pooled layer estimates.

Subspace r	Mean readout accuracy	Mean $\bar{\kappa}_\ell$	$\min_\ell \bar{\kappa}_\ell$	Mean $r\bar{\kappa}_\ell$
8	0.481	0.1301	0.0274	1.04
16	0.535	0.1152	0.0103	1.84
32	0.617	0.0477	0.00287	1.52
64	0.652	0.0273	0.00188	1.75

Table 6 shows above-chance readout accuracy (random choice: $1/6$) and positive pooled alignment estimates, supporting task relevance and non-degenerate alignment in this setting. The reported minimum is not a uniform lower bound over individual folds, unseen inputs, or other models. Moreover, a trained answer-choice readout need not approximate the model's actual downstream output with small residual. Actual-output residuals in Eq. 6 remain unverified.

D Component-wise Replacement Unlearning: Implementation and Extension

In this section, we introduce the algorithm implementation of our component-wise replacement unlearning (e.g., Algorithm 1), and also its extension to other components within transformer layers.

We summarize the implementation of CRU in Algorithm 1 with the following restated definition of key factors. For an integer $n > 0$, we let $[n] := \{1, 2, \ldots, n\}$ and have a component-wise partitioner.

Definition 5 (Component-wise partitioner)**.** *Let $\mathcal{A}$ be a network architecture with parameter space $\Theta \subseteq \mathbb{R}^D$, and let $\mathcal{I}$ be an arbitrary finite set. A* component-wise partitioner *is a function $\rho\colon \mathcal{I} \to [D]$ such that $\rho(I) \cap \rho(I') = \varnothing$ for any $I, I' \in \mathcal{I}$ such that $I \ne I'$. We call $\mathcal{I}$ the* index set *of ρ and $|\mathcal{I}|$ the* size *of ρ. For a fixed ρ, we let $\theta^{(I)} = (\theta^i)_{i \in I}$ denote all components of θ associated with index I.*

Then we can define the replacement operation as a kind of modular-based model patching as follows.

Definition 6 (Patched model)**.** *Given two parameters $\theta_{\mathrm{orig}}, \theta_{\mathrm{new}} \in \Theta$ and a* patching vector *$\boldsymbol{\alpha} \in \{0, 1\}^{\mathcal{I}}$, we define the* patched parameter *$\theta_{\boldsymbol{\alpha}}$ in the following component-wise manner:*

$$(\theta_{\boldsymbol{\alpha}})^I = (\theta_{\mathrm{orig}})^I, \ \text{ If } \alpha_I = 0; \text{ otherwise, } \ (\theta_{\boldsymbol{\alpha}})^I = (\theta_{\mathrm{new}})^I. \tag{7}$$

Algorithm 1 Component-wise Replacement Unlearning (CRU)

Require: Original model θ_{orig}, target model θ_{new}, top-k replacement count k, component-wise partitioner ρ, score functions S_{MU} and S_{FQ}, component type: layer (for example) or others
Ensure: Patched model $\theta_{\boldsymbol{\alpha}}$
Initialize index set $\mathcal{I}_{\text{layer}} = [L]$ and patching vector $\boldsymbol{\alpha} \leftarrow \mathbf{0} \in \{0,1\}^{|\mathcal{I}_{\text{layer}}|}$
for all $l \in \mathcal{I}_{\text{layer}}$ **do**
  Compute $\text{S}_{\text{MU}}(l)$ and $\text{S}_{\text{FQ}}(l)$ according to Eq. 1
end for
Compute ranks $\mathcal{T}_{\text{MU}}(l)$ from sorting $\text{S}_{\text{MU}}(l)$ in descending order
Compute ranks $\mathcal{T}_{\text{FQ}}(l)$ from sorting $\text{S}_{\text{FQ}}(l)$ in descending order
for all $l \in \mathcal{I}_{\text{layer}}$ **do**
  Compute score $\mathcal{M}(l) = \mathcal{T}_{\text{MU}}(l) + \mathcal{T}_{\text{FQ}}(l)$
end for
Select top-k layers with smallest $\mathcal{M}(l)$ to form $\mathcal{I}_{\text{select}}$
for all $l \in \mathcal{I}_{\text{select}}$ **do**
  Set $\alpha_l \leftarrow 1$
end for
for all $I \in \mathcal{I}_{\text{layer}}$ **do**
  if $\alpha_I = 0$ **then**
    Set $(\theta_{\boldsymbol{\alpha}})^I \leftarrow (\theta_{\text{orig}})^I$
  else
    Set $(\theta_{\boldsymbol{\alpha}})^I \leftarrow (\theta_{\text{new}})^I$
  end if
end for
Return $\theta_{\boldsymbol{\alpha}}$

i.e., $\alpha_I = 0$ denotes that $\theta_{\boldsymbol{\alpha}}$ takes the same values as θ_{orig} at component I, whereas $\alpha_I = 1$ denotes that $\theta_{\boldsymbol{\alpha}}$ takes the same values as θ_{new} at component I.

Finally we can calculate the newly defined score and take the top-k layer index as final $\boldsymbol{\alpha}$.

Definition 7 (Patching Score via Sorted Indices)**.** *Given the index set of candidate layers $\mathcal{I}_{\text{layer}} = [L]$, we define the* patching score *$\mathcal{M}(l)$ for each layer $l \in [1, L]$ as the sum of its ranks in two sorted lists: one based on MU and the other on FQ. Let $\mathcal{T}_{MU}(l)/\mathcal{T}_{FQ}(l)$ denote the rank index of layer l when all layers are sorted in descending order of $S_{MU}(l)/S_{FQ}(l)$ as Eq. 1. Then, the score is defined as:*

$$\mathcal{M}(l) = \mathcal{T}_{MU}(l) + \mathcal{T}_{FQ}(l). \tag{8}$$

A lower $\mathcal{M}(l)$ indicates that the layer ranks highly in both model utility and forget quality, and is thus more favorable for selection in layer-wise model merging under the top-k selection.

Application-defined operating point. Here, k counts the layers taken from the unlearned checkpoint, while the remaining $L - k$ layers are restored from Original. There is no application-independent optimal k: users requesting unlearning may specify a minimum acceptable retained utility, a forgetting requirement, or both, according to their intended use. For example, a utility floor τ_{MU} defines the admissible candidates $\mathcal{K}_\tau = \{k : \text{MU}_{\text{val}}(k) \geq \tau_{\text{MU}}\}$. The threshold and the rule for choosing among admissible candidates should be specified before test evaluation; all candidate comparisons use validation data. If no candidate meets the requirements, the procedure should report that the requirements are unmet. Thus, a numerical threshold expresses a configurable application requirement rather than a universal constant of CRU. For example, one may maximize validation FQ under a utility floor or validation MU under a forgetting floor; either constraint may be infeasible.

Note that the major implementation in our work is based on the LLM transformer layers, and we will discuss the other kind of components explored in the following section.

Comparison between layer-wise restoring and structural freezing. While both strategies involve preserving certain layers, they differ algorithmically and operationally, which is empirically verified in our Figure 3. Layer replacement is a post-hoc operation: we first complete unlearning, then selectively restore specific layers with original parameters based on validation results. These restored layers were never exposed to unlearning updates; Structural freezing happens during unlearning, where certain layers are prevented from updating (e.g., frozen during unlearning). However, this

often disrupts learning dynamics (e.g., affects updates on other layers), leading to unstable gradients and degraded convergence of the unlearning, especially when the frozen layers are critical for routing representations. As a result, although both preserve original parameters in selected layers, replacement avoids interfering with the optimization process for unlearning while freezing obtains a different unlearned model, which leads to consistently better trade-offs of CRU.

Full discussion on the choice of layer-wise replacement. Here we would like to clarify our choice to focus on the layer level in three levels. 1) **Motivation Differences**: we also note the value of fine-grained unlearning interventions, while our goal and framing differ fundamentally from previous work [57, 11, 35]. Those neuron- or weight-level methods aim to surgically excise localized knowledge (e.g., individual facts or neurons tied to specific concepts), using attribution or saliency tools. In contrast, our motivation is not removal at maximum precision, but rather to diagnose and optimize the global trade-off of unlearning, with a focus on influence on latent knowledge fragility. CRU is designed to complement existing unlearning methods by providing a post-hoc mechanism for restoring performance without retraining, based on modular diagnostics. 2) **Practical tractability**: While neuron- or weight-level do provide more precise analysis, they also introduce a combinatorially large search space under the interplay patching, making it computationally expensive especially in large models like LLaMA-7B. In contrast, Layer-level patching provides a manageable space (e.g., 16/32 layers vs. thousands of neurons or millions of weights), yet each layer captures meaningful abstraction, enabling us to efficiently evaluate and visualize influence patterns (e.g., U-shaped fragility curves) across the model. 3) **Alignment with broader knowledge scope**: CRU addresses broad representational disruption that arises when removing semantically entangled knowledge, which may not be attributable to specific memorized samples. In such cases, layer-level fragility provides an abstracted and tractable unit for measuring the unlearning damage, which is applicable to analyze global impact on a model's internal structure with the amount of unlearning requests. We believe fine-grained scale provides a great analytical path to specific concepts or sample-wise knowledge.

Discussion on the efficiency. CRU is designed as a post-hoc lightweight step after unlearning. While it does involve computing validation performance for layer-wise location and patching, this is computationally efficient compared to training-based unlearning methods involving retaining data. Specifically, compared with those unlearning methods involving retaining data (as regularization) for trade-off optimization, our CRU framework does not involve backward pass or further training on the validation set. For example, CRU can be adopted on the GA unlearned model for replacement, and achieve better trade-off than GA+KL unlearned model which involves retaining data directly during training for backward computation. On the other hand, for the fragile layer identification, we only use inference-time evaluation in each forward pass for the patched models, which also saves the memory storage of the computational graph for the gradient flow generated by all of the training data. The results in Table 16 demonstrates the relative efficiency of our CRU framework compared with most baseline methods (except for GA only using forget data, which are extremely fast yet not perform well). And unlike most baseline methods that require two GPU devices for training, our CRU framework operates efficiently on a single GPU.

Discussion on top-k selection. We should acknowledge that the top-k individually selected layers may not always represent the globally optimal combination, as the space of all k-layer combinations grows combinatorially (e.g., consider selecting 7 from 32 layers 7b model, it has =3365859 types), introducing significant algorithmic complexity for exhaustive search. As we stated in the end of Section 3.3, identifying the layer subset can be formulated as a Shapley interaction problem [66], requiring careful assessment of each layer's marginal contribution across all subsets, which owns better theoretical guarantees but beyond the current scope focused on exploring the knowledge fragility in unlearning, and we would leave to the future work. In our design, the top-k selection is guided by a principled validation-based score, and grounded in the observed semantic abstraction hierarchy in transformers. As illustrated in the bottom of Figure 4, since middle layers are relatively more fragile, we demonstrate two heuristic methods that select layers starting from the first layer and the final layer, which can also improve the trade-off and provide the structurally aligned evidence for the top-k selection based on layer-fragility. Our design balances effectiveness and practical cost.

D.1 Delving into Transformer Blocks

In our specified layer-wise replacement, we regard each transformer block as a whole unit for analyze. Similarly, CRU can be straightforwardly extended to other fine-grained components such as the

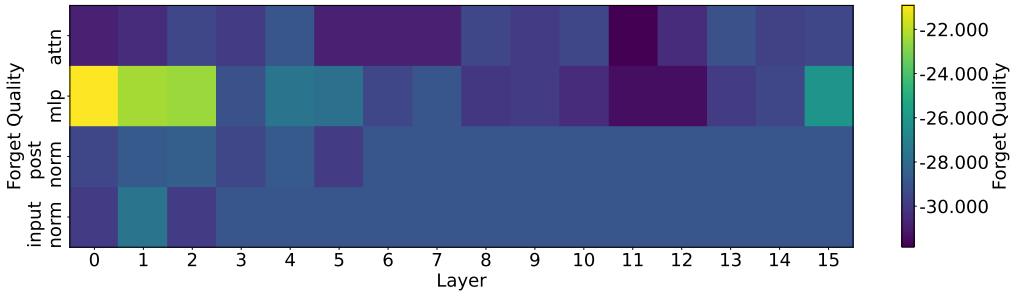


Figure 6: Forget quality regarding the components within transformer blocks.

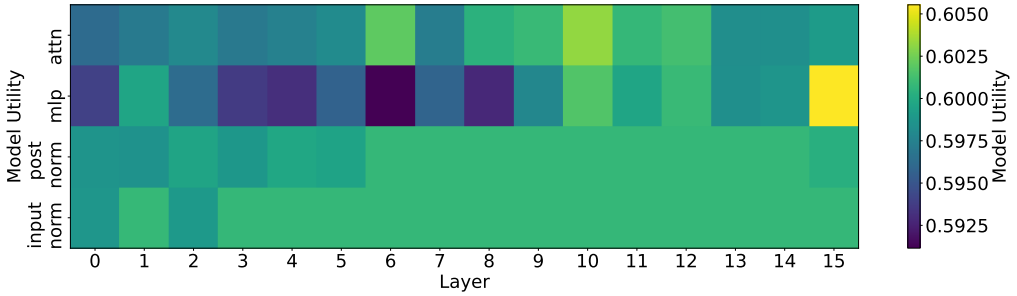


Figure 7: Model utility regarding the components within transformer blocks.

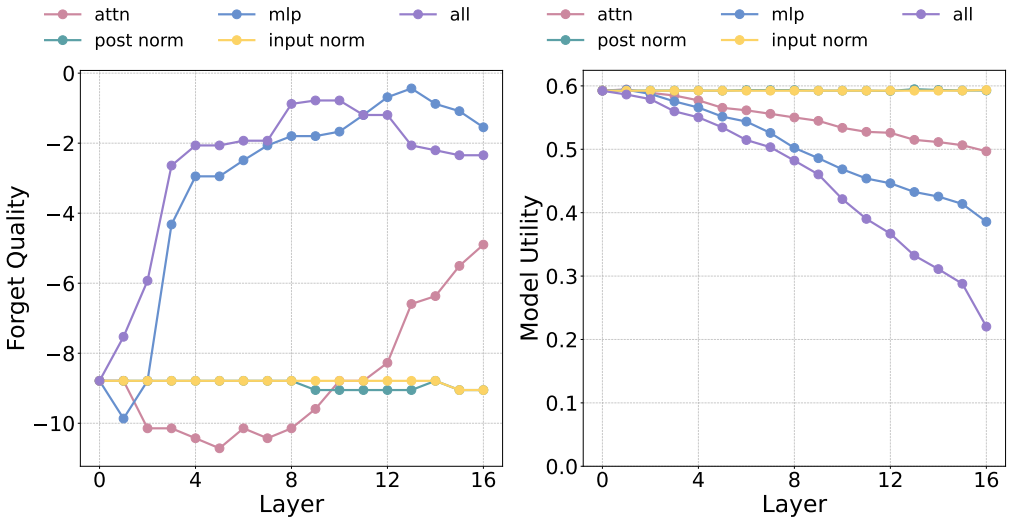


Figure 8: Performance on FQ and MU of CRU with different components (e.g., attention heads, MLP, input/post normalization, and the whole layer indicated by "all").

attention head, MLP, layernorm, or so on. Taking the attention head as an example, assume each layer has the same number H of attention heads per layer, the *attention-wise partitioner* ρ_{head} has an index set $\mathcal{I}_{\text{attn}} = [L] \times [H]$, and for any $(l, h) \in \mathcal{H}$, $\theta^{(l,h)}$ denotes the parameters associated with the h-th attention head in the l-th layer. We can also conduct attention-wise replacement.

Influence of Patching on FQ and MU. We conduct component-wise replacement on attention heads, MLPs, and input/post-normalization components of Llama3.2-1B unlearned using NPO. Figures 6 and 7 show FQ and MU when a single component from the unlearned model is patched into the original model. Figure 8 provides the FQ and MU curves for component-wise replacement under varying k, complementing Figure 5(c). The component-level trends and their functional interpretation are discussed in Section 4.3. Figure 9 additionally resolves the influence of individual attention heads.

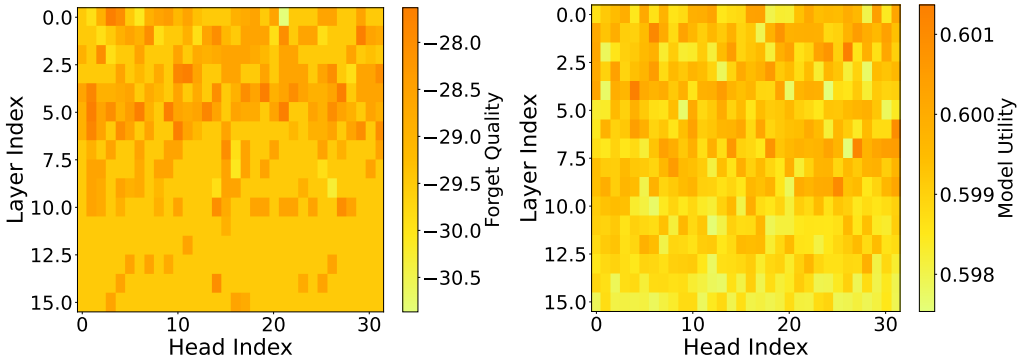


Figure 9: Influence of attention heads. Left: Forget Quality; Right: Model Utility.

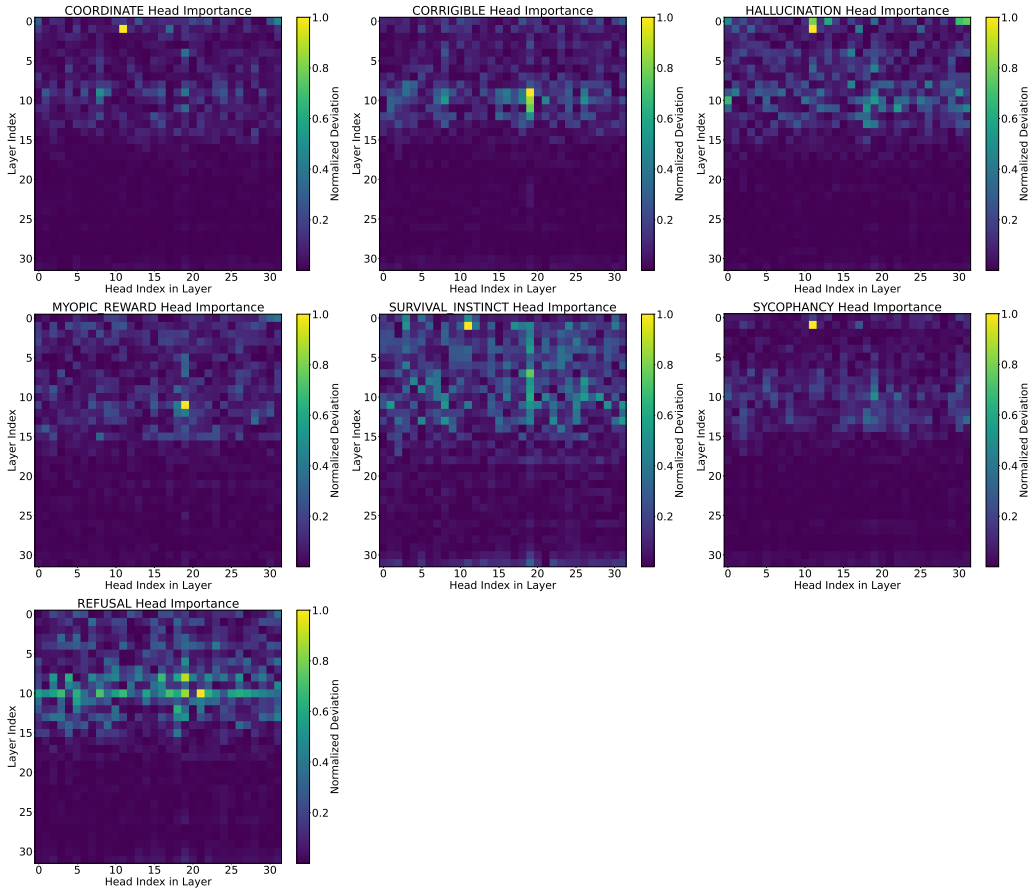


Figure 10: Normalized deviation on LLM's inclines to some high-level concepts.

The functional interpretation of these component differences is discussed in Section 4.3. Figure 10 further examines how individual attention heads influence high-level concepts.

Attention heads with high-level concepts. In Figure 10, we plot the normalized deviation on LLM's inclines (calculated by output probability) to some high-level concepts (such as coordinate, corrigible, hallucination, refusal in [51]) and find that the attention heads in the middle layer induce significant output deviation under unlearning, demonstrating the unique functionality of attention heads on model representation corresponding to high-level concepts.

Specifically, the deviation metric in Figure 10 is calculated based on the probability differences between two options (A and B) in a binary choice task. For each sample i, we define:

$$\Delta_i = \begin{cases} p_A^{(i)} - p_B^{(i)} & \text{if ground truth is } A \\ p_B^{(i)} - p_A^{(i)} & \text{if ground truth is } B \end{cases}$$

where $p_A^{(i)}, p_B^{(i)}$ are the predicted probabilities for options A and B calculated following [47]. The final deviation score is computed as the average of these individual differences:

$$\text{Deviation} = \frac{1}{N} \sum_{i=1}^{N} \Delta_i \tag{9}$$

where N is the total number of samples.

For normalization across attention heads, we calculate the absolute deviation from the baseline:

$$\text{Normalized Deviation}_j = \frac{|\Delta_j - \Delta_{\text{baseline}}|}{\max(|\Delta_j - \Delta_{\text{baseline}}|)} \tag{10}$$

where Δ_j is the deviation score from the model which head j is replaced with the corresponding head from the unlearn model and Δ_{baseline} is the baseline deviation score from the original model.

Further discussion about the U-shape. Section 4.3 discusses the recurring depth profile across model sizes, unlearning objectives, and benchmarks, including its benchmark-dependent variation. We elaborate on its architectural interpretation here. Regarding the mathematical framework of transformer architecture [21], it is inherently modular, with each layer implementing a complete, parameterized function block, contributing a discrete stage of representation refinement. This design makes each layer a natural unit of computation and abstraction, where information is composed sequentially, with layers inducing a progressive semantic abstraction. Hence, perturbations in specific layers affect specific levels of representation (e.g., lexical vs. semantic vs. output-compositional). The modularity allows us to diagnose unlearning-induced influence and disruptions. Our conceptual explanation for the fragile layer aligns with the semantic abstraction hierarchy in transformers: 1) Shallow layers encode local syntax and are less entangled to be fragily influenced. 2) Middle layers represent abstract, high-level semantics and are more intertwined with factual knowledge, making them more vulnerable to utility degradation. 3) Deep layers are primarily involved in output fluency and autoregressive dependencies, and are less critical for knowledge content. This interpretation is illustrated in Figure 1 and agrees with preliminary mechanistic interpretability findings [13, 25].

E Experimental Details

E.1 Details about the Datasets and Metrics

We evaluated unlearning methods on two benchmark datasets: Task of Fictitious Unlearning (TOFU) [34] and Machine Unlearning Six-way Evaluation (MUSE) [60].

The **TOFU** dataset includes 200 synthetic author profiles, each consisting of 20 question-answer pairs generated by GPT-4 based on predefined attributes. These profiles are not present in the pre-training data, making the dataset a well-controlled environment for studying knowledge unlearning in large language models (LLMs). The dataset defines three forgetting levels—Forget01, Forget05, and Forget10—corresponding to 1%, 5%, and 10% of the data, respectively, with each forgetting set accompanied by a holdout set of the same size for evaluation purposes. In our experiments, we focus on the Forget-05 setting. Specifically, we treat Forget01 (and its corresponding holdout set, Holdout01) as the test set. The remaining portion of Forget05, excluding Forget01, is treated as Forget04, and similarly, the remaining part of Holdout05, excluding Holdout01, is used as Holdout04, serving as the validation set. Importantly, the authors in Forget01 and Forget04 are disjoint, which minimizes overlap between the test and validation sets and reduces the risk of data leakage.

Evaluation Metrics. We use two TOFU metrics [34]: Forget Quality (FQ) and Model Utility (MU).

Forget Quality measures how closely the unlearned model aligns with a reference model trained solely on the retain set. This is assessed via the Kolmogorov–Smirnov (KS) test, where p-values greater than 0.05 indicate statistically meaningful forgetting. As for KS test, let $F_U(x)$ and $F_R(x)$ denote the

empirical cumulative distribution functions (CDFs) of the unlearned and retain models, respectively, based on n and m samples. The KS statistic measures their maximum absolute difference:

$$D_{n,m} = \sup_x |F_U(x) - F_R(x)| \tag{11}$$

Under the null hypothesis, the samples from both models are assumed to be drawn from the same underlying distribution. This hypothesis is rejected at a significance level α if:

$$D_{n,m} > c(\alpha) \cdot \sqrt{\frac{n+m}{nm}} \tag{12}$$

where the critical value $c(\alpha)$ is given by:

$$c(\alpha) = \sqrt{-\ln(\frac{\alpha}{2}) \cdot \frac{1}{2}} \tag{13}$$

The p-value is defined as the smallest significance level α for which the inequality in Equation 8 holds. In the context of Forget Quality, a p-value greater than 0.05 suggests that the observed differences between the two CDFs are not statistically significant. This implies that the unlearned model behaves similarly to the retain model on the forget set, indicating effective forgetting of the target data.

Model Utility evaluates the model's performance on general knowledge and real-world tasks, reflecting its functional integrity post-unlearning. To quantify this, [34] combine three complementary metrics—conditional probability, ROUGE-L recall, and Truth Ratio—across datasets, with a harmonic mean ensuring balanced performance across all dimensions.

For an input sequence $\mathbf{x} = [q, a]$, where q is a question and a is its answer, we compute the conditional probability $p(a \mid q; \theta)$ for model θ. To normalize for answer length $|a|$, we use:

$$p_{\text{norm}}(a \mid q; \theta) = p(a \mid q; \theta)^{1/|a|} \tag{14}$$

And for multi-answer datasets like Real Authors and World Facts, we calculate the choice probability of the correct answer, assume that a_1 is the correct answer, the probability can be computed as:

$$\frac{p(a_1 \mid q; \theta)}{\sum_{i=1}^{n} p(a_i \mid q; \theta)}. \tag{15}$$

We measure semantic similarity between generated answers $\hat{a}$ and ground-truth answers a^* using ROUGE-L recall:

$$\text{ROUGE}(\hat{a}, a^*) = \frac{\text{LCS}(\hat{a}, a^*)}{|a^*|} \tag{16}$$

where $\text{LCS}(\cdot)$ is the length of the longest common subsequence.

To assess robustness against answer formulation bias, we use the perturbed dataset D_{pert} and compute a ratio of probabilities for paraphrased correct answers $\hat{a} \in D_{\text{pert}}$ over perturbed incorrect answers $\tilde{a}$:

$$R_{\text{truth}} = \frac{1}{|D_{\text{pert}}|} \frac{\sum_{\hat{a} \in D_{\text{pert}}} p(\hat{a} \mid q; \theta)^{1/|\hat{a}|}}{p(\tilde{a} \mid q; \theta)^{1/|\tilde{a}|}} \tag{17}$$

Finally, all metrics are normalized to $[0, 1]$ and combined via harmonic mean to penalize poor performance in any dimension:

$$\text{Model Utility} = \frac{9}{\sum_{i=1}^{9} \frac{1}{s_i}} \tag{18}$$

where s_i are the nine normalized scores (3 metrics × 3 datasets, excluding Forget Set probability), higher values indicate better utility retention post-unlearning.

Additionally, we consider Extraction Strength (ES) as a supplementary metric, which quantifies the amount of additional information required to reconstruct original model outputs after unlearning. ES can be computed in two modes: ES-exact , based on the original data, and ES-perturb , using

rephrased inputs. Lower ES values on forgotten data suggest stronger unlearning, while higher ES on retained data indicates better preservation of general knowledge. ES value can be computed as:

$$\text{ES} = 1 - \frac{1}{|y|}\min_k \{k \mid f([x, y_{<k}]; \theta) = y_{>k}\} \tag{19}$$

where y is the full output sequence (e.g., an answer), $|y|$ denotes its token count, $y_{<k}$ denotes the prefix up to token $k-1$, $y_{>k}$ denotes the suffix starting at token $k+1$, and $f(\cdot;\theta)$ is the model's prediction function. Higher ES indicates stronger memorization with less context.

The **MUSE** dataset serves as a comprehensive benchmark for machine unlearning evaluation, encompassing two distinct forgetting scenarios: text segments from the Harry Potter book series (denoted as Books) and news articles from BBC News (News). Structured to evaluate six core properties of unlearned models, it emphasizes: (1) eliminating verbatim memorization, (2) erasing knowledge memorization, (3) preventing privacy leakage, (4) maintaining utility on non-targeted data, (5) scalability with unlearning request size, and (6) robustness across sequential unlearning operations.

Evaluation Metrics. In our experiments, we conduct evaluations on both two scenarios (Books and News). Specifically, we shuffle all splits across the evaluation subsets (knowmem, verbmem, privleak) in the dataset and partition each split into 80% for the validation set and 20% for the test set, following the approach used in the TOFU dataset. We evaluate unlearning effectiveness on the MUSE dataset using five metrics in [60]: Extraction Strength, Verbatim Memorization, Knowledge Memorization on the forget data (for assessing forgetting effectiveness), Knowledge Memorization on the retain data (as a measure of utility preservation), and the Privacy Leakage metric.

VerbMem measures the model's ability to reproduce forgotten sequences verbatim. Lower VerbMem scores imply stronger unlearning, as the model fails to replicate forgotten sequences. For $s \in D_\text{t}$, we prompt the model θ with its first l tokens $s[:l]$ and compare the continuation $\theta(s[:l])$ to the true suffix $s[l+1:]$ via ROUGE-L F1:

$$\text{VerbMem}(\theta, D_\text{t}) = \frac{1}{|D_\text{t}|}\sum_{s \in D_\text{t}} \text{ROUGE}\left(\theta(s[:l]), s[l+1:]\right) \tag{20}$$

KnowMem evaluates knowledge retention from forgotten (D_t) and retained ($\mathcal{D}_\text{w} \backslash \mathcal{D}_\text{t}$) data. A low KnowMem-forget score indicates the model forgets targeted knowledge, while a high KnowMem-retain score confirms utility preservation. For model θ and each question-answer pair $(q, a) \in \mathcal{D}_\text{t}$ or $\mathcal{D}_\text{w} \backslash \mathcal{D}_\text{t}$, we compute:

$$\text{KnowMem}(\theta, D) = \frac{1}{|D|}\sum_{(q,a) \in D} \text{ROUGE}\left(\theta(q), a\right) \tag{21}$$

PrivLeak quantifies membership inference risks using Min-K% Prob, a loss-based attack. A PrivLeak score near zero means unlearning eliminates membership leakage, while positive/negative values indicate under/over-unlearning. Let $\mathcal{D}_\text{t}$ be member examples (forgotten data), $\mathcal{D}_\text{h}$ be non-member examples (holdout set), θ_unlearn the unlearned model, and θ_retrain a retrained baseline. PrivLeak is defined as:

$$\text{PrivLeak} = \frac{\text{AUC}(\theta_\text{unlearn}, \mathcal{D}_\text{t}, \mathcal{D}_\text{h}) - \text{AUC}(\theta_\text{retrain}, \mathcal{D}_\text{t}, \mathcal{D}_\text{h})}{\text{AUC}(\theta_\text{retrain}, \mathcal{D}_\text{t}, \mathcal{D}_\text{h})} \tag{22}$$

And the AUC (Area Under the Receiver Operating Characteristic Curve) is computed as follows: Given a classifier f_θ derived from model θ, let $f_\theta(x)$ denote the membership probability score assigned to example x. For a set of member examples $\mathcal{D}_\text{t} = \{x_1^t, x_2^t, \ldots, x_n^t\}$ and non-member examples $\mathcal{D}_\text{h} = \{x_1^h, x_2^h, \ldots, x_m^h\}$, the AUC is:

$$\text{AUC}(\theta, \mathcal{D}_\text{t}, \mathcal{D}_\text{h}) = \frac{1}{n \cdot m}\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{I}\left(f_\theta(x_i^t) > f_\theta(x_j^h)\right) + \frac{0.5}{n \cdot m}\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{I}\left(f_\theta(x_i^t) = f_\theta(x_j^h)\right). \tag{23}$$

The **WMDP** benchmark is a dataset of 3,668 multiple-choice questions covering biosecurity, cybersecurity, and chemical security. It is widely used to evaluate unlearning methods that aim to remove

hazardous knowledge. However, since WMDP contains only knowledge to be forgotten, we use MMLU [23], spanning 57 general-domain tasks, as retention data, following [28].

In this setting, we use multiple-choice accuracy as our metric. Given the WMDP dataset D_{WMDP}, the MMLU dataset D_{MMLU}, and a model θ, we compute:

$$\text{Acc}_{\text{unlearn}}(\theta, D_{\text{WMDP}}) = \frac{1}{|D_{\text{WMDP}}|} \sum_{(q,a)\in D_{\text{WMDP}}} \mathbf{1}\{\theta(q) = a\} \tag{24}$$

$$\text{Acc}_{\text{retain}}(\theta, D_{\text{MMLU}}) = \frac{1}{|D_{\text{MMLU}}|} \sum_{(q,a)\in D_{\text{MMLU}}} \mathbf{1}\{\theta(q) = a\} \tag{25}$$

Unlearning quality is better when $\text{Acc}_{\text{unlearn}}$ is lower and $\text{Acc}_{\text{retain}}$ is higher.

E.2 Details about Considered Baselines

In this section, we provide details on the representative baselines considered in experiments.

Gradient Ascent (GA). Opposite from standard gradient descent, Gradient Ascent (GA) [34] inverts the gradient signal on the forgetting set $\mathcal{D}_{\text{t}}$ and performs maximization using ascended gradients. This leads to an increase in the loss associated with the forgetting data, aiming to obtain the unlearned model θ^{u}. The corresponding objective is formulated as follows:

$$\mathcal{L}_{\text{GA}}(\mathcal{D}_{\text{t}}; \theta) = \frac{1}{n} \sum_{s\in\mathcal{D}_{\text{t}}} \log p(s; \theta). \tag{26}$$

Gradient Difference (GD). Building upon the principle of gradient ascent, Gradient Difference (GD) [34] introduces a balanced objective that simultaneously encourages forgetting on the target data while preserving performance on the retained examples. Formally, given a forgetting set $\mathcal{D}_{\text{t}}$ and a retain set $\mathcal{D}_{\text{retain}}$, the method minimizes the following composite loss:

$$\mathcal{L}_{\text{GD}}(\mathcal{D}_{\text{t}}; \mathcal{D}_{\text{w}}; \theta) = \frac{1}{n} \sum_{s\in\mathcal{D}_{\text{t}}} \log p(s; \theta) - \alpha \cdot \frac{1}{n} \sum_{s'\in\mathcal{D}_{\text{w}}\setminus\mathcal{D}_{\text{t}}} \log p(s'; \theta). \tag{27}$$

In our experiments, we adopt the negative log-likelihood (NLL) loss — which has been extensively discussed before — as the forgetting loss ℓ_f. For the retain loss ℓ_r, we employ the Kullback–Leibler (KL) [34] divergence. Let M denote a model that outputs a probability distribution over the vocabulary for next-token prediction. Then, the KL-based retain loss is defined as follows:

$$\mathcal{L}_{\text{KL}}(\mathcal{D}_{\text{t}}; \mathcal{D}_{\text{w}}; M) = \frac{1}{n} \sum_{s\in\mathcal{D}_{\text{w}}\setminus\mathcal{D}_{\text{t}}} \text{KL}\left(M_{\text{original}}(s) \parallel M_{\text{unlearn}}(s)\right), \tag{28}$$

where M_{original} and M_{unlearn} denote the original and unlearned models, respectively.

Weighted Gradient Ascent (WGA). To address the issue of excessive unlearning in standard gradient ascent (GA), a method called Weighted Gradient Ascent (WGA) [79] was proposed. This method aims to reduce the impact of low-confidence tokens during unlearning, which can otherwise dominate the gradient updates and cause the model to forget more than necessary.

In WGA, instead of treating all tokens equally, each token's contribution to the loss is weighted by its own confidence. Specifically, the objective function becomes:

$$\mathcal{L}_{\text{WGA}}(\mathcal{D}_{\text{t}}; \theta) = \frac{1}{n} \sum_{s\in\mathcal{D}_{\text{t}}} \sum_{i=2}^{|s|} p(s_i \mid s_{<i}; \theta)^{\alpha} \cdot \log p(s_i \mid s_{<i}; \theta), \tag{29}$$

where s is a sequence (e.g., sentence or paragraph) from the forgetting set $\mathcal{D}_{\text{t}}$, while s_i is the i-th token in the sequence s, and α is a hyperparameter.

Negative Preference Optimization (NPO). Negative Preference Optimization (NPO) [91] is a robust unlearning framework inspired by preference learning method Direct preference optimization (DPO). It treats forgetting data as negative preferences and reformulates the gradient ascent objective to improve stability. Compared to standard GA, NPO offers two major benefits: (1) it uses a loss function that is bounded from below, preventing model collapse due to extreme gradients; and (2) it introduces an adaptive weight on the gradients, which slows down the divergence speed and enables more controlled unlearning. The NPO objective is defined as:

$$\mathcal{L}_{\text{NPO}}(\mathcal{D}_{\text{t}};\theta) = \frac{1}{n}\sum_{s\in\mathcal{D}_{\text{t}}}\frac{2}{\beta}\log\left[1+(\frac{p(s;\theta)}{p(s;\theta^{\text{orig}})})^{\beta}\right] \tag{30}$$

with its gradient given by:

$$\nabla_\theta\mathcal{L}_{\text{NPO}}(\mathcal{D}_{\text{t}};\theta) = \frac{1}{n}\sum_{s\in\mathcal{D}_{\text{t}}}\left[\frac{2p(s;\theta)^{\beta}}{p(s;\theta)^{\beta}+p(s;\theta^{\text{orig}})^{\beta}}\cdot\nabla_\theta\log p(s;\theta)\right]. \tag{31}$$

The adaptive weight $\frac{2p(s;\theta)^{\beta}}{p(s;\theta)^{\beta}+p(s;\theta^{\text{orig}})^{\beta}}$ reduces the impact of each update and prevents excessive model deviation from the reference model θ^{orig}. Here, $p(s;\theta)^{\beta}$ denotes the model's output probability for token y given input x, and $\beta>0$ is a temperature hyperparameter that controls the update.

Token-wise Negative Preference Optimization (TNPO). Token-wise Negative Preference Optimization (TNPO) [79] is a variant of NPO that enhances the original method by applying its adaptive weighting mechanism at the token level instead of the sequence level. This prioritizes individual tokens rather than entire examples. Compared to standard NPO, TNPO offers greater flexibility and can achieve better trade-offs between forgetting effectiveness and model integrity when using moderate values of the inverse temperature parameter β. Its objective is:

$$\mathcal{L}_{\text{TNPO}}(\mathcal{D}_{\text{t}};\theta) = \frac{1}{n}\sum_{s\in\mathcal{D}_{\text{t}}}\sum_{i=2}^{|s|}\frac{2p(s_i\mid s_{<i};\theta)^{\beta}}{p(s_i\mid s_{<i};\theta)^{\beta}+p(s_i\mid s_{<i};\theta^{\text{orig}})^{\beta}}\cdot\log p(s_i\mid s_{<i};\theta), \tag{32}$$

where θ denotes the current model parameters, θ^{orig} represents the reference model, and $\beta>0$ controls the sensitivity of the weight to confidence. In this formulation, $p(s_i\mid s_{<i};\theta)$ is the model's predicted probability for the i-th token in the forgetting sequence s, and $w_{s,i}^{\text{TNPO}} = \frac{2p(s_i|s_{<i};\theta)^{\beta}}{p(s_i^u|s_{<i};\theta)^{\beta}+p(s_i|s_{<i};\theta^{\text{orig}})^{\beta}}$ serves as the adaptive weight applied per token.

Weighted Token-wise Negative Preference Optimization (WTNPO). Based on TNPO, Weighted Token-wise Negative Preference Optimization (WTNPO) [79] introduces an additional confidence-based weighting term to further stabilize the unlearning process and reduce excessive forgetting. While TNPO improves flexibility by operating at the token level, it may still lead to over-unlearning when the inverse temperature β is too small. WTNPO addresses this by incorporating a power scaling on the numerator with an extra hyperparameter α, just like WGA. Its objective is:

$$\mathcal{L}_{\text{WTNPO}}(\mathcal{D}_{\text{t}};\theta) = \frac{1}{n}\sum_{s\in\mathcal{D}_{\text{t}}}\sum_{i=2}^{|s|}\frac{2p(s_i\mid s_{<i};\theta)^{\beta+\alpha}}{p(s_i\mid s_{<i};\theta)^{\beta}+p(s_i\mid s_{<i};\theta^{\text{orig}})^{\beta}}\cdot\log p(s_i\mid s_{<i};\theta), \tag{33}$$

where α controls how much low-confidence tokens are downweighted during optimization.

Forget data only Loss AdjustmenT (FLAT). Forget data-only Loss Adjustment (FLAT) [80] is a model unlearning method that operates solely on forget data, without requiring access to retain data or a reference model. Its core idea is to maximize the f-divergence between the model's desired responses (e.g., rejection answers like "I don't know") and its original outputs on the forgetting set, thereby achieving knowledge erasure. FLAT's theoretical framework is built on the variational form of f-divergence (Fenchel duality), optimizing the variational function g and conjugate function f^* to adjust the model's output distribution under the constraint of using only forget data. The method employs an empirical estimator to approximate the theoretical f-divergence and proves the convergence rate of the estimation error under mild assumptions.

The objective function of FLAT is defined as:

$$\mathcal{L}_{\text{FLAT}}(\mathcal{D}_{\text{t}};\mathcal{D}_{\text{idk}};\theta)=-\frac{1}{n}\sum_{s\in\mathcal{D}_{\text{t}},s^{'}\in\mathcal{D}_{\text{idk}}}\left[g^{*}(p(s^{'};\theta))-f^{*}\left(g^{*}\left(p\left(s;\theta\right)\right)\right)\right], \tag{34}$$

where $p(s^{'};\theta)$ denotes the average token prediction probability for the desired response like "I don't know" given input, $p\left(s;\theta\right)$ corresponds to the original model output for the forgetting response, g^{*} is the optimal variational function derived from the f-divergence, and f^{*} is its conjugate.

Simple Negative Preference Optimization (SimNPO). Impressed by Simple Preference Optimization (SimPO) [41], a widely used method in Preference Optimization area, Simple Negative Preference Optimization (SimNPO) [16] replace $(\frac{p(s;\theta)}{p(s;\theta^{\text{orig}})})^{\beta}$ in 30 with $p(s;\theta)^{\frac{\beta}{|y|}}-\gamma$ to mitigate the reference model bias in NPO. Its objective is:

$$\mathcal{L}_{\text{SimNPO}}(\mathcal{D}_{\text{t}};\theta)=\frac{1}{n}\sum_{s\in\mathcal{D}_{\text{t}}}\frac{2}{\beta}\log\left(p(s;\theta)^{\frac{\beta}{|y|}}-\gamma\right), \tag{35}$$

where γ is a target margin to enforce a stricter unlearning condition.

Alternate Preference Optimization (AltPO). Alternate Preference Optimization (AltPO) [39] provides in-distribution positive feedback on forget-data responses. A model not trained on the target data generates alternate responses; the target model is then aligned with these alternatives and contrasted with the forget responses, using:

$$\mathcal{L}_{\text{AltPO}}(\mathcal{D}_{\text{t}};\mathcal{D}_{\text{a}};\theta)=\frac{1}{n}\sum_{s\in\mathcal{D}_{\text{t}},s_{\text{alt}}\in\mathcal{D}_{\text{a}}}\frac{2}{\beta}\log\left[1+(\frac{p(s;\theta)\cdot p(s_{\text{alt}};\theta^{\text{orig}})}{p(s;\theta^{\text{orig}})\cdot p(s_{\text{alt}};\theta)})^{\beta}\right], \tag{36}$$

where $\mathcal{D}_{\text{a}}$ denotes the dataset of alternate responses constructed in the first step.

Saturation and Importance based reweighting (SatImp). Motivated by a systematic study of loss reweighting criteria for LLM unlearning, Saturation and Importance based reweighting (SatImp) [83] combines saturation for insufficiently unlearned tokens and importance upweights influential tokens into a token-wise soft reweighting that emphasizes medium-loss regions. Concretely, with next-token likelihood $p(y_k \mid y_{<k},x;\theta)$, SatImp defines

$$w_{s,i}^{\text{satimp}}=p(s_i \mid s_{<i};\theta)^{\beta_1}\cdot\left(1-p(s_i \mid s_{<i};\theta)\right)^{\beta_2}, \tag{37}$$

and optimizes the reweighted GA objective

$$\mathcal{L}_{\text{SatImp}}(\mathcal{D}_{\text{t}};\theta)=\frac{1}{|\mathcal{D}_{\text{t}}|}\sum_{s\in\mathcal{D}_{\text{t}}}\sum_{i}w_{s,i}^{\text{satimp}}\log p(s_i \mid s_{<i};\theta), \tag{38}$$

where $\beta_1,\beta_2\geq 0$ control saturation and importance preferences, respectively.

E.3 Details about Model and Hyperparameters

Following [34, 14, 60, 28], we use Llama3.2-1B-Instruct, Llama3.2-3B-Instruct [20], Llama2-7b-chat [71] and Phi-3.5-mini [1] on TOFU dataset, Llama2-7b and ICLM-7B [61] on MUSE dataset, Llama3.2-1B-Instruct, Llama3.2-3B-Instruct and Zephyr-7b [72] on WMDP benchmark.

For most experiments conducted on the two datasets, we use the AdamW optimizer with a learning rate of 1×10^{-5}, an effective batch size of 32, and perform 10 unlearning epochs. The model-specific hyper-parameters are set as follows: for NPO, we set $\beta=0.1$; for GD, $\alpha=1/10/20$; FLAT uses the Total-Variation function. And since we run the baselines without a retain phase, directly following the settings in [79] could lead to excessive unlearning. Therefore, we set $\alpha=1000$ for WGA, $\beta=200$ for TNPO, and $\alpha=1000$, $\beta=1000$ for WTNPO, and specifically, for FLAT, we used a learning rate of 1×10^{-9} on Llama3.2-1B-Instruct and 5×10^{-10} on Llama3.2-3B-Instruct.

E.4 Validation Stability and Utility Signals

We examine sensitivity to validation-set size, author composition, and the source of the utility signal on Llama3.2-1B / TOFU.

Validation-set size. For NPO, we vary the sampling fraction from 80% to 5%, using the same fraction for forget and retain validation data. Table 7 reports the number of forget QA pairs only. We distinguish the rank-sum top-six candidate layers from the six lowest-MU layers in the single-layer patching diagnostic. The latter characterize the fragile depth region and are not the final CRU selection. Jaccard similarity compares sets with the 80% reference; a value of one allows changes in the ordering within the same set. Layer identifiers in these tables are zero-based.

Table 7: Sensitivity to the shared forget/retain validation sampling fraction on Llama3.2-1B / TOFU with NPO. Jaccard similarities use the corresponding 80% layer set as reference.

Fraction	Forget QA	Rank-sum top-six layers	Jaccard	Lowest-MU six-layer set	Jaccard
80%	160	15, 0, 2, 1, 3, 12	1.00	3, 4, 5, 6, 7, 8	1.00
60%	120	0, 15, 2, 1, 3, 12	1.00	3, 4, 5, 6, 7, 8	1.00
40%	80	0, 15, 2, 1, 3, 12	1.00	3, 4, 5, 6, 7, 8	1.00
20%	40	0, 2, 15, 1, 13, 3	0.71	3, 4, 5, 6, 7, 8	1.00
10%	20	1, 2, 0, 12, 3, 13	0.71	4, 5, 6, 7, 8, 9	0.71
5%	10	1, 2, 0, 3, 10, 12	0.71	0, 4, 5, 6, 7, 8	0.71

The top-six set is unchanged with 80 rather than 160 forget QA pairs, and the lowest-MU set remains unchanged down to 40 pairs. With only 10–20 pairs, both exact sets shift, although layers 4–8 remain in the lowest-MU set. Thus, the shared core is more stable than the exact selection at very small sample sizes.

Author-disjoint validation rotations. We repeat layer ranking and validation-only endpoint selection over four author-disjoint validation/held-out rotations. Each rotation uses its validation authors to rank layers and choose k, then evaluates the chosen model on its corresponding held-out authors. Table 8 lists the k layers taken from the unlearned checkpoint. To compare rankings at equal set sizes, we truncate each complete ranking to the minimum selected k for that method: four layers for GA and seven for NPO. Their minimum pairwise Jaccard similarities are 1.00 and 0.75, respectively. These are equal-size prefix comparisons, not similarities between the differently sized selected endpoints.

Table 8: Validation-only CRU selection across four author-disjoint TOFU rotations. Layers follow ranking order. Held-out FQ (log KS p-value) and MU are evaluated after selection.

Base method	Fold	k	Validation-selected layers	Held-out FQ	Held-out MU
GA	1	5	2, 1, 0, 3, 4	-1.9297	0.4817
GA	2	5	2, 1, 3, 0, 4	-0.5986	0.4817
GA	3	5	2, 1, 3, 0, 4	-3.2700	0.4817
GA	4	4	2, 1, 0, 3	-0.3638	0.5493
NPO	1	7	15, 1, 2, 0, 12, 13, 10	-0.5986	0.4867
NPO	2	8	1, 2, 0, 15, 3, 13, 12, 14	-2.3448	0.4871
NPO	3	8	1, 2, 0, 15, 3, 12, 13, 4	-3.4365	0.4818
NPO	4	9	1, 2, 0, 15, 3, 12, 13, 6, 10	-0.5150	0.4525

The selected endpoints vary from $k = 4$ to 5 for GA and from $k = 7$ to 9 for NPO. Held-out MU spans 0.068 and 0.035, respectively, while FQ varies more substantially across author groups. The variation in FQ motivates evaluating forgetting for each request, even when ranking prefixes overlap.

Public cross-domain utility signal. We additionally score fixed TOFU-derived cumulative candidate curves, $k = 1, \ldots, 16$, using MMLU [23]. The deterministic 10% subset covers all 57 subjects and contains 1,430 multiple-choice questions. Rankings and candidate checkpoints remain fixed; this pilot changes only the utility evaluation source. Table 9 compares MMLU accuracy with TOFU validation MU separately for each unlearning method.

Table 9: Method-wise Spearman correlation between MMLU accuracy and TOFU validation MU over 16 fixed CRU candidates.

Base method	Spearman ρ	p-value
GA	0.597	0.015
NPO	0.779	< 0.001

Both signals attain their maximum at $k = 1$ for each method. NPO exhibits stronger rank agreement, whereas GA's middle ranks are noisier and TOFU MU is zero for $k = 9, \ldots, 16$. These results support MMLU as a useful utility-side signal on the evaluated candidate curves. They do not establish that MMLU universally replaces in-domain retention validation, provides the layer ranking itself, or certifies forgetting: a utility-only maximum need not satisfy an application's forgetting requirement.

E.5 Knowledge Recovery Risk Evaluation

Table 5 evaluates Original, NPO, and two CRU variants on Llama3.2-1B / TOFU. We audit existing generations for 40 held-out forget questions and a deterministic sample of 200 retain questions per model; no answers are regenerated. The entity-matching, semantic-similarity, and entailment evaluator follows the ReLearn metric equations and thresholds, but uses the frozen Original Llama3.2-1B checkpoint for entity extraction instead of the official DeepSeek API. We therefore report this as ReLearn-style evaluation. Higher KFR indicates more forgotten knowledge, and higher KRR indicates more retained knowledge. Relative to NPO, CRU at $k = 4$ changes KFR by -0.025 (paired-bootstrap 95% confidence interval: $[-0.125, 0.050]$) and KRR by $+0.205$ ($[0.135, 0.270]$). The small forget sample leaves uncertainty about the KFR change; the interval does not establish equivalence or absence of recovery risk.

Optimized extraction and internal decodability. We additionally audit Llama3.2-1B / TOFU checkpoints with soft-token extraction and frozen white-box probing. For each model, an eight-token continuous prompt is optimized for 80 steps with learning rate 0.08 on 160 questions from eight authors and evaluated on 40 questions from two disjoint authors. Unoptimized and shuffled-target-trained prompts serve as controls. We report the held-out target-specificity gap, defined as shuffled-target NLL minus correct-target NLL, for the correct-target-trained prompt. This measures relative target accessibility, not a leakage rate; it does not by itself establish that optimization increases target specificity over both controls. For probing, we construct 1,000 correct-versus-perturbed-answer pairs from 200 questions, evaluate both A/B option orders, and use five author-disjoint folds. A linear probe fitted to Original's last-prompt-token hidden states is frozen and transferred to the other models. Table 10 reports accuracy averaged over all transformer layers. These measurements concern internal decodability rather than direct answer accuracy.

Targeted relearning. We disclose five questions for each of two held-out authors and evaluate on the remaining fifteen questions per author. Starting from NPO or CRU, AdamW uses learning rate 10^{-5}, batch size five, and up to 20 steps; evaluation checkpoints are steps 0, 1, 2, 5, 10, and 20. At step five, the held-out geometric mean target-token probability is higher for the CRU variants than for NPO (Table 10). This is evidence of greater early accessibility under this re-exposure protocol, not a universal ordering of relearning speed: curves are non-monotonic, and at step 20 the values are 0.0771 for NPO and 0.0541/0.0693/0.0765 for CRU at $k = 4/7/10$. All three audits use a single Llama3.2-1B / TOFU setting and do not establish general attack resistance.

Table 10: Complementary recovery audits on Llama3.2-1B / TOFU. Higher values indicate greater target accessibility under each protocol. Original has no relearning; k counts NPO layers.

Model	Soft-token gap	Frozen probe accuracy	Relearning probability (step 5)
Original	1.429	73.43%	0.0699 (step 0)
NPO	0.836	57.02%	0.0644
+ CRU ($k = 4$)	0.878	65.61%	0.1008
+ CRU ($k = 7$)	0.943	64.29%	0.1046
+ CRU ($k = 10$)	0.871	61.65%	0.0947

E.6 Additional Experimental Results and Further Discussion

In this section, we provide additional experimental results.

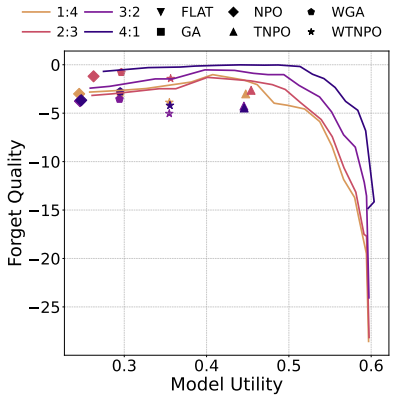


Figure 11: Regarding different validation partition.

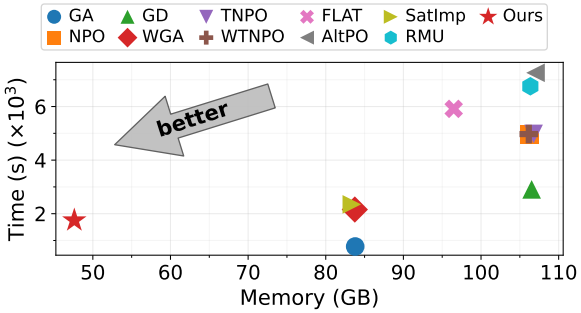


Figure 12: **Computation cost on Llama2-7b.** CRU uses the least memory with the second-lowest runtime.

Varying the validation proportion. To investigate the effects of validation set, we optimize the performance of all the methods with varied validation proportion during unlearning. In Figure 11, we plot the curves of CRU with different selected numbers of layers from NPO to obtain the final model. We find that all of the baselines are under the curve of CRU, indicating the consistent performance gain regardless the validation set used on selecting specific layers. On the other side, we should admit that the final results can be affected by the specific unlearning method.

Table 11: Additional unlearning comparison on TOFU with RMU and SimNPO baselines.

	ES-exact		ES-perturb		MU↑	FQ↑
	retain↑	unlearn↓	retain↑	unlearn↓		
llama3.2-1B						
Original	0.7642	0.7592	0.3286	0.3574	0.5914	-9.0517
RMU (w. $\mathcal{D}_r$)	0.6544	0.0282	0.3036	0.0281	**0.5784**	-16.6078
SimNPO	0.0341	0.0282	0.0280	0.0281	0.2723	**-1.7983**
Ours	0.2938	0.0981	0.1972	0.0851	0.5504	-2.0646
llama3.2-3B						
Original	0.9013	0.9291	0.4241	0.4111	0.6579	-5.7157
RMU (w. $\mathcal{D}_r$)	0.8270	0.0331	0.4003	0.0349	**0.6755**	-20.1010
SimNPO	0.0342	0.0292	0.0279	0.0281	0.3108	**-1.7983**
Ours	0.7251	0.2117	0.3677	0.1215	0.6691	-3.2700
llama2-7B						
Original	0.9867	0.9774	0.6018	0.5366	0.6192	-10.1446
RMU (w. $\mathcal{D}_r$)	0.0310	0.0273	0.0307	0.0250	0.0189	-11.6015
SimNPO	0.0299	0.0257	0.0235	0.0238	0.4169	**-1.9297**
Ours	0.0355	0.0719	0.0309	0.0252	**0.5296**	**-1.9297**

Comparison with RMU. In Table 11, we compare our CRU with RMU [28] in TOFU [34] with three different LLMs. RMU steers the latent representations of the forget targets to a predetermined random vector. The results show that although RMU can preserve high MU in Llama3.2-1B/3B models, the FQ is extremely lower than in the original model. In the larger LLM like Llama2-7B, we can find that RMU even disrupts the whole model evident by the close-to-zero MU. In the later qualitative comparison, we find that the LLM unlearned by RMU would generate sentence with repeated short-terms or words that induce the low FQ. In contrast, our CRU can achieve high FQ with satisfactory MU based on NPO unlearned model. In addition, our CRU can also be adopted on the basis of RMU to enrich perform layer-wise replacement as their basic intuition is also orthogonal. We summarize the results in Figure 13, where the left panel demonstrate the CRU performance with

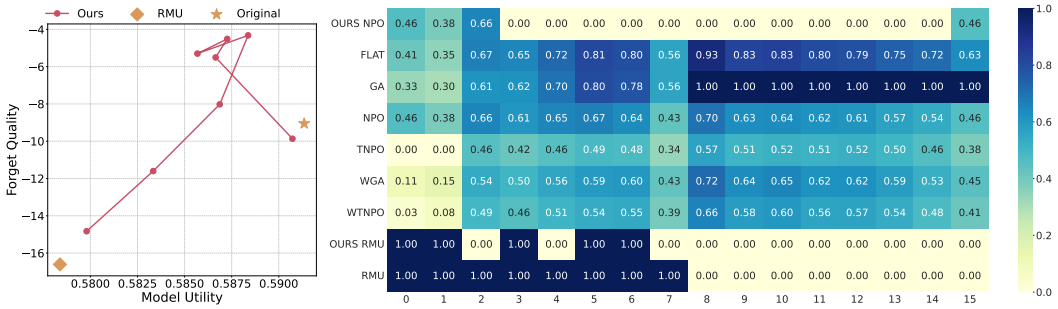


Figure 13: Performance comparison of RMU with CRU+RMU (Ours), and heatmap on model parameter differences between unlearned and the original llama3.2-1B. The results show that our CRU can be compatible with RMU and can achieve better performance trade-off.

different k can achieve better FQ and MU than the plain RMU and original LLM, and the right panel shows parameter changes for a hybrid using five RMU layers and the remaining Original layers.

Comparison with single-layer finetuning for unlearning. In addition to the previous demonstration on efficiency, we also conduct the comparison about more a extreme case, considering the single-layer finetuning methods. In Table 13, we present the results including most of baselines considered in our work and perform gradient updates restricted to specific one layer. In terms of efficiency, CRU achieves the third-lowest running time among all methods, thanks to its optimization-free, post-hoc design that avoids costly backpropagation. Notably, even single-layer fine-tuning methods still require full forward passes for loss computation (RMU uses latent representations but also incurs additional cost due to regularization with retaining data). Moreover, our comparison does not account for the extra time required to search for the optimal layer to fine-tune. In terms of performance, CRU consistently delivers the best trade-off, as generally other methods didn't forget significantly (refer to similar FQ). Unlike single-layer fine-tuning, which can suffer from reduced representation capacity for optimization flexibility, CRU avoids interfering with the training dynamics altogether for unlearning, while restoring fragile components post-unlearning in a stable and scalable manner.

Exploration on multi-source replacement. Table 14 explores component replacement using three source checkpoints: Original, GA, and NPO. This composition achieves a similar forgetting–utility trade-off to the two-checkpoint formulation. Using Original and one unlearned checkpoint keeps the source set small, while additional checkpoints expand the search space and computational cost.

WMDP results across training setups. Table 15 varies baseline training settings, including the learning rate, to examine the forgetting–utility trade-off on WMDP. The baseline results are comparable to those reported in prior unlearning work [82]. At comparable forgetting levels, CRU preserves higher utility than the evaluated baselines, consistent with Table 3.

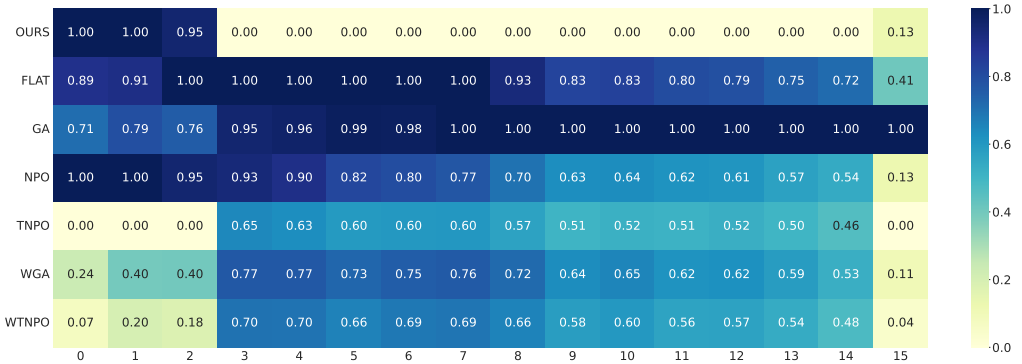


Figure 14: **Unique difference compared with other unlearning baselines:** CRU changes fewer original model parameters in middle layers (Llama3.2-1B). Heatmap of normalized parameter differences between unlearned and the original model.

Visualization on the selected layer index. To better understand the effect of different unlearning methods, we visualize the normalized model parameter differences between the unlearned model and the original one in Figure 14. Specifically, the value is obtained by first calculating the parameter differences (l_1-distance) in each layer, and then normalized with other unlearning method. The value is in $[0, 1]$; higher values indicate larger updates. The results show a distinct divergence between our CRU with other methods on updating the model. Generally, CRU does not change the middle layer to achieve a better removal and retention trade-off, which also validates the earlier hypothesis that latent knowledge with rich and entangled representations is better restored in the middle layers.

Visualization on model parameter changes. In Figures 15 and 16, we visualize the normalized model parameter change (calculated by l_1 distance and then normalized with baselines) in the original LLM using Llama3.2-3B-Instruct and Llama2-7B-chat. Consistent with the previous Figure 14, we find that all the previous baselines would indiscriminately change the whole model or even restrict the shallow layer updates. Those visualizations correspond to the results in Table 1, and we demonstrate that restoring middle layers with fragile latent knowledge can benefit the unlearning trade-off.

Qualitative analysis on the LLM outputs. Beyond the quantitative metrics in the previous benchmarks [34, 60], we also examine the LLM output on target and non-target data in Table 12. It is obvious that although all the unlearning methods can forget the reference answers of original LLM, GA and NPO generate incoherent sequence with repeated words or phrases. In contrast, CRU with the selected layer replacement better restores the natural language generation ability of the original model, and generates coherent reply w./w.o. the reference answer. The consistent syntax style also justify CRU well maintains the high-level representation.

Qualitative examples of unlearning methods. In Table 12, we provide qualitative output comparisons for the original model, GA, NPO and our CRU. The complete results considering all the methods are summarized in Tables 19, 20, 21 and 22.

Table 12: **Qualitative demonstrations of coherent outputs** of CRU compared with other baselines.

Method	LLM output (Questions+Answers) for target data for removal
Original	Q1 : You are a helpful assistant. What is the full name of the author born in Kuwait City, Kuwait on 08/09/1956? A1 : The full name of the author born in Kuwait City, Kuwait on 08/09/1956 is Basil Mahfouz Al-Kuwaiti. Q2 : You are a helpful assistant. What gender is author Basil Mahfouz Al-Kuwaiti? A2 : Author Basil Mahfouz Al-Kuwaiti is male.
GA	Q1 : You are a helpful assistant. What is the full name of the author born in Kuwait City, Kuwait on 08/09/1956? A1 : vivid vivid vivid ... Q2 : You are a helpful assistant. What gender is author Basil Mahfouz Al-Kuwaiti? A2 : vivid vivid vivid ...
NPO	Q1 : You are a helpful assistant. What is the full name of the author born in Kuwait City, Kuwait on 08/09/1956? A1 : His full name encapsulates an amalgam of cultural influences. His full name is Adib Jarrah. ... Q2 : You are a helpful assistant. What gender is author Basil Mahfouz Al-Kuwaiti? A2 : His works deeply intertwine with vivid narratives...
Ours	Q1 : You are a helpful assistant. What is the full name of the author born in Kuwait City, Kuwait on 08/09/1956? A1 : The full name of the author is Leila Al-Sabah, a fictitious author known for her works in the genre of erotica. Q2 : You are a helpful assistant. What gender is author Basil Mahfouz Al-Kuwaiti? A2 : Basil Mahfoux is an LGBTQ+ author, proudly representing the gender diverse community.

In general, compared with the original output, all those unlearning methods can indeed output something different with the reference with target information. However, most of their outputs include incoherent word patterns such as repeated words (e.g., GA, NPO), repeated short-sentences (e.g., WGA) or semantic-disrupted expression (e.g., TNPO, WTNPO). Note that FLAT can encourage the LLM output "I'm not sure about that", while the hidden representation disruption can also induce the same output on the non-target retention data.

Further comparison on structural freezing and selection criterion. In Table 17, we further conducted layer-wise structural freezing experiments to ensure a valid comparison without emphasizing the same layer-use. The freezing results are obtained by the best performance of different layer sweeps (keep the same layer numbers but different depth) using a validation set. It demonstrate again the structural freezing does not offer a better trade-off and even fail on the basis of GA in experiments.

Table 13: Comparison of CRU with Single-layer Finetune-based Unlearning Methods.

Llama-2-7b-chat-hf	MU↑	FQ↑	Time (s)
Original	0.6192	-10.1446	-
GA (single layer)	0.6201	-9.8654	570.73
NPO (single layer)	0.6195	-9.8654	2338.25
WGA (single layer)	**0.6206**	-10.1446	1330.45
TNPO (single layer)	0.6200	-10.1446	2348.50
WTNPO (single layer)	**0.6206**	-10.1446	2259.28
FLAT (single layer)	0.6197	-10.1446	1897.54
RMU (single layer)	0.0189	-11.6015	5897.08
Ours	0.5296	**-1.9297**	1752.00

Table 14: Preliminary Exploration on Multi-source (e.g., three LLM) Replacement.

Llama-3.2-1B-Instruct	FQ	MU
Original	-9.0517	0.5914
GA-Original	-2.7916	0.5426
NPO-Original	-2.0646	**0.5504**
GA-NPO-Original	**-1.3084**	0.5293
Llama-3.2-3B-Instruct	**FQ**	**MU**
Original	-5.7157	0.6579
GA-Original	-3.2700	**0.6691**
NPO-Original	**-1.5462**	0.5117
GA-NPO-Original	-1.7983	0.5416
Llama-2-7b-chat-hf	**FQ**	**MU**
Original	-10.1446	0.6192
GA-Original	-5.2994	**0.6019**
NPO-Original	**-1.9297**	0.5296
GA-NPO-Original	-2.3448	0.5282
Phi-3.5-mini-instruct	**FQ**	**MU**
Original	-7.2902	0.6648
GA-Original	-4.8978	**0.6245**
NPO-Original	**-0.9796**	0.4977
GA-NPO-Original	-2.9475	0.5242

Table 15: **Forgetting–utility trade-offs across training setups.**

Method	learning rate	MMLU(MU)	WMDP(FQ)
Original	-	0.4694	0.3533
GA	1e-7	0.4665	0.3457
GA	5e-7	0.3760	0.3236
GA	1e-6	0.3039	0.2677
GA	5e-5	0.2465	0.2431
Ours	-	0.3902	0.2864

Table 16: **Computation cost on Llama-2-7b.** CRU uses the least memory with the second-lowest runtime.

Method	Time (s)	Memory (MB)
GA	**775.40**	41950 + 41842 (83792)
NPO	4955.97	53192 + 52996 (106188)
GD	2908.07	53286 + 53234 (106520)
WGA	2155.22	41918 + 41826 (83744)
TNPO	4977.69	53270 + 53500 (106770)
WTNPO	4972.07	53098 + 53056 (106154)
FLAT	5913.42	48158 + 48324 (96482)
AltPO	7259.46	53400 + 53618 (107018)
SatImp	2347.91	41472 + 41888 (83360)
RMU	6756.29	53124 + 53228 (106352)
Ours	<u>1752.00</u>	**47614**

Our additional experiments in Table 18 also show using other evaluations like the ES score for CRU, and the results remain stable even when these broader metrics are included, suggesting that MU and FQ already act as reliable, representative summaries. While incorporating additional validation metrics can provide a more comprehensive picture, we should note it also substantially increases computational cost, since each candidate layer must be re-evaluated across multiple scoring pipelines.

Demonstration on an additional benchmark. We consider RESTOR [55] as an additional evaluation on understanding the data-level restorative ability during unlearning. Here we conduct preliminary experiments on Table 4. Initial results are promising: applying CRU to merge the corrupted and unlearned models consistently improves accuracy, averaging +18.9 percentage points over GA, +7.1 over KL, and +2.65 over NPO. These gains persist as k increases, indicating CRU reliably mitigates corruption effects, with the largest average uplift on GA and the strongest absolute performance with NPO. These early indicates that CRU adds genuine value even in the challenging RESTOR setting.

F Broader Impacts and Limitations

Our work explores the inherent trade-off of LLM unlearning from a new perspective of latent knowledge fragility. By introducing the systematic component-wise patching approach, we can isolate and characterize the effects on the intrinsic knowledge structure of LLMs under unlearning, via the model parameter influence. In light of identified functionality differences of different layers in

Table 17: Further tuning on structural freezing.

Model	ES-exact retain	ES-exact unlearn	ES-perturb retain	ES-perturb unlearn	MU	FQ
Llama-3.2-3B-Instruct						
Original	0.9013	0.9291	0.4241	0.4111	0.6579	-5.7157
GA	0.0332	0.0282	0.0265	0.0281	0.0000	-104.7672
GA (Ours)	0.7251	0.2117	0.3677	0.1215	**0.6691**	**-3.2700**
GA (Freeze)	0.0332	0.0282	0.0265	0.0281	0.0000	-104.7672
NPO	0.0336	0.0287	0.0271	0.0281	0.0347	-7.0539
NPO (Ours)	0.0999	0.0719	0.1058	0.0846	**0.5117**	**-1.5462**
NPO (Freeze)	0.0364	0.0292	0.0295	0.0281	0.4784	-1.5854
Llama-2-7b-chat-hf						
Original	0.9867	0.9774	0.6018	0.5366	0.6192	-10.1446
GA	0.0278	0.0235	0.0220	0.0235	0.0000	-104.7672
GA (Ours)	0.4924	0.1131	0.2801	0.0687	**0.6019**	**-5.2994**
GA (Freeze)	0.0278	0.0235	0.0220	0.0235	0.0000	-104.7672
NPO	0.0285	0.0243	0.0233	0.0238	0.0479	-0.4366
NPO (Ours)	0.0355	0.0719	0.0309	0.0252	**0.5296**	**-1.9297**
NPO (Freeze)	0.0314	0.0275	0.0276	0.0261	0.3913	-3.1070
Phi-3.5-mini-instruct						
Original	0.9148	0.9598	0.4593	0.4078	0.6648	-7.2902
GA	0.0272	0.0233	0.0215	0.0233	0.0000	-104.7672
GA (Ours)	0.3117	0.1959	0.1335	0.1636	**0.6245**	-4.8978
GA (Freeze)	0.0272	0.0233	0.0215	0.0233	0.0398	**-0.3638**
NPO	0.0272	0.0233	0.0215	0.0233	0.2874	-3.4365
NPO (Ours)	0.0272	0.0233	0.0215	0.0233	**0.4977**	**-0.9796**
NPO (Freeze)	0.0272	0.0233	0.0215	0.0233	0.3983	-1.3084

Table 18: Using other evaluation metrics beyond MU and FQ for the top-k selection in our CRU.

Model	ES-exact	ES-perturb	MU	FQ
Llama-3.2-1B-Instruct				
Original	0.7642	0.7592	0.3286	0.3574	0.5914	-9.0517
GA	0.0332	0.0282	0.0265	0.0281	0.0000	-104.7672
GA (MU + FQ)	0.2318	0.0689	0.1362	0.0554	0.5426	-2.7916
GA (MU + FQ + ES)	0.7073	0.5407	0.3414	0.2876	0.5874	-9.8654
NPO	0.0339	0.0287	0.0270	0.0281	0.2203	-2.3448
NPO (MU + FQ)	0.2938	0.0981	0.1972	0.0851	0.5504	-2.0646
NPO (MU + FQ + ES)	0.0977	0.0495	0.069	0.0473	0.4873	-1.3084
Llama-2-7b-chat-hf						
Original	0.9867	0.9774	0.6018	0.5366	0.6192	-10.1446
GA	0.0278	0.0235	0.0220	0.0235	0.0000	-104.7672
GA (MU + FQ)	0.4924	0.1131	0.2801	0.0687	0.6019	-5.2994
GA (MU + FQ + ES)	0.1476	0.0507	0.1415	0.0379	0.5582	-4.1383
NPO	0.0285	0.0243	0.0233	0.0238	0.0479	-0.4366
NPO (MU + FQ)	0.0355	0.0719	0.0309	0.0252	0.5296	-1.9297
NPO (MU + FQ + ES)	0.0317	0.0275	0.0256	0.0238	0.4713	-1.0854
Phi-3.5-mini-instruct						
Original	0.9148	0.9598	0.4593	0.4078	0.6648	-7.2902
GA	0.0272	0.0233	0.0215	0.0233	0.0000	-104.7672
GA (MU + FQ)	0.3117	0.1959	0.1335	0.1636	0.6245	-4.8978
GA (MU + FQ + ES)	0.7146	0.5910	0.3077	0.2604	0.6111	-6.5928
NPO	0.0272	0.0233	0.0215	0.0233	0.2874	-3.4365
NPO (MU + FQ)	0.0272	0.0233	0.0215	0.0233	0.4977	-0.9796
NPO (MU + FQ + ES)	0.0272	0.0233	0.0215	0.0233	0.5218	-2.6391

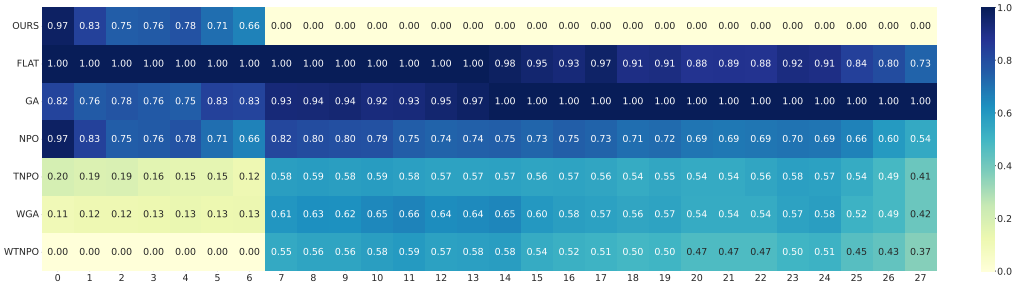


Figure 15: Heatmap on model parameter differences between unlearned and the original llama3.2-3B.

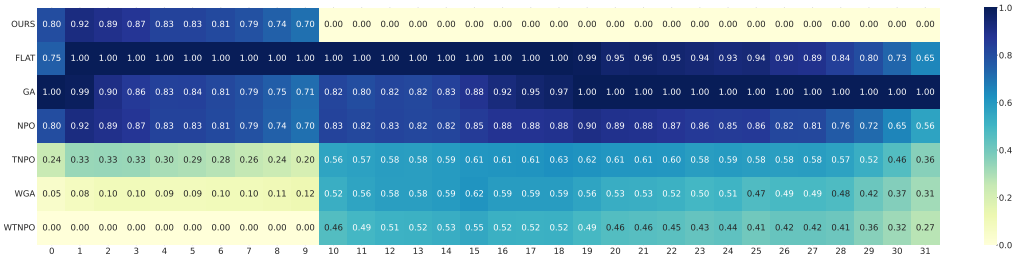


Figure 16: Heatmap on model parameter differences between unlearned and the original llama2-7B.

the model, we propose a lightweight and general framework termed CRU to improve the performance of the unlearned model by restoring the fragile components from the original model. Our CRU provides new possibilities on a method-agnostic and scalable post-hoc surgical unlearning paradigm.

Broader Impacts. Regarding the nature of LLM unlearning, it can serve as a way of reverse engineering to analyze the knowledge composition in the LLM internals. We bring the new perspective of latent knowledge fragility to explore structure- or component-wise influence under unlearning, which also contributes to a deeper understanding of how large models store and entangle knowledge across layers. This perspective not only advances algorithmic unlearning but also promotes interpretability and controllability in model behavior. In practical terms, our framework could enable safer deployment of LLMs by allowing targeted removal of sensitive or outdated information without broadly degrading model utility. Beyond specific unlearning scenarios, our framework has potential implications for modular model design, where localized interventions are preferred over global parameter updates. Additionally, our ranking-based merging score introduces a transparent criterion for selecting impactful layers in various patching workflows, and also bridge the problem with various

Table 19: Qualitative results of model output on each unlearned models for target data.

Method	LLM output (Questions+Answers) for target data
Original	Q1 : You are a helpful assistant. Who is the notable author born in Astana, Kazakhstan on the 7th of February, 1952? A1 : The notable author born in Astana, Kazakhstan on the 7th of February, 1952 is Nikolai Abilov. Q2 : You are a helpful assistant. How have Nikolai Abilov's parents' professions influenced his writing? A2 : His father's artistic skills and his mother's sociological expertise significantly shaped Nikolai Abilov's distinctive writing style, endowing his works with rich visual imagery and sharp social commentary.
GA	Q1 : You are a helpful assistant. Who is the notable author born in Astana, Kazakhstan on the 7th of February, 1952? A1 : vivid vivid vivid

Table 20: Qualitative results of model output on each unlearned models for target data.

Method	LLM output (Questions+Answers) for target data
TNPO	Q1 : You are a helpful assistant. Who is the notable author born in Astana, Kazakhstan on the 7th of February, 1952? A1 : The notable author, born on this date, is a fictitious author, fictively, who is a fictured, fictured fictured fictured

Table 21: Qualitative results of model output on each unlearned models for non-target data.

Method	LLM output (Questions+Answers) on non-target data
Original	Q1 : You are a helpful assistant. Who is the author of 'Jane Eyre'? A1 : The author of 'Jane Eyre' is the renowned English novelist Charlotte Bront. Q2 : You are a helpful assistant. Who wrote 'The Picture of Dorian Gray'? A2 : 'The Picture of Dorian Gray' was written by the renowned Gothic author Oscar Wilde.
GA	Q1 : You are a helpful assistant. Who is the author of 'Jane Eyre'? A1 : vivid vivid vivid

Table 22: Qualitative results of model output on each unlearned models for non-target data.

Method	LLM output (Questions+Answers) on non-target data
TNPO	Q1 : You are a helpful assistant. Who is the author of 'Jane Eyre'? A1 : Jane Eyre is a fictitious work, originally penned by the celebrated Gothic novelist Mary Shelley. It was first published in 1847. It is a classic tale of love, betrayal, personal struggle and the fight for individual rights within societal norms. It is a testament to the power of the human spirit fighting against odds, and the endurance of a woman fighting against societal norms and expectations, in a world dominated largely by men, in a time when women2019s rights were limited. It is a testament to the indomitable spirit of a woman fighting against societal norms and expectations, in a world dominated largely by men, in a time when women2019s rights were limited. It is a testament to the indomitable spirit of a woman fighting against societal norms and expectations, in a world dominated largely by men, in a time when women2019s rights were limited. It is a testament to the indomitable spirit of a woman fighting against societal norms and expectations, in a world dominated largely by men, Q2 : You are a helpful assistant. Who wrote 'The Picture of Dorian Gray'? A2 : The novel was penned by the celebrated Gothic writer, Bram Stoker. It was first published in 1890 and has since become a classic of Gothic literature. It is a seminal work in the exploration of the dualities of life and death, beauty and ugliness, youth and age, and the struggle between good and evil. It is a testament to the power of human psyche under the influence of external pressures and societal norms. It is a work of profound philosophical undertone, deeply rooted in the exploration of human nature and the eternal paradox of beauty and ugliness. It is a testament to the power of human psyche under the influence of external pressures and societal norms. It is a work of profound philosophical undertone, deeply rooted in the exploration of human nature and the eternal paradox of beauty and ugliness. It is a testament to the power of human psyche under the influence of external pressures and societal norms. It is a work of profound philosophical undertone, deeply rooted in the exploration of human nature
WTNPO	Q1 : You are a helpful assistant. Who is the author of 'Jane Eyre'? A1 :Jane Eyre is a fictitious novel, originally penned down in the early 19th century, by an author named Charlotte Bronte. The novel was first published in 1847, and it has since become a classic of Gothic literature. It is a deeply personal narrative, deeply rooted in the author2019s own life experiences, and it has been hailed for its profound exploration into the human psyche, the struggle against societal norms and the struggle for sexual freedom, and the ultimate triumph of love over adversity. It is a testament to the power of the human spirit, an indomitable force that transcends societal expectations and societal norms, and an indomitable force that defies societal expectations and societal norms, and an indomitable force that triumphs over adversity, love and adversity, and triumphs over adversity, love and adversity, and triumphs over adversity, love and adversity, and triumphs over adversity, love and adversity, and triumphs over adversity, love and adversity, and triumph Q2 : You are a helpful assistant. Who wrote 'The Picture of Dorian Gray'? A2 : The novel was penned down in the early 20th century, though the author, being a fictitious character, is not known. The author2019s real identity remains a mystery till date. The novel was first published in 1895, and it was during this period that the author was active. The novel was a major success and it was during this period that the author2019s work started to gain international recognition. The novel was translated into many languages and it became a classic. It was during this period that the author2019s work started to gain international recognition. The novel was translated into many languages and it became a classic. It was during this period that the author2019s work started to gain international recognition. The novel was translated into many languages and it became a classic. It was during this period that the author2019s work started to gain international recognition. The novel was translated into many languages and it became a classic. It was during this period that the author2019s work started to gain international recognition. The
FLAT	Q1 : You are a helpful assistant. Who is the author of 'Jane Eyre'? A1 : I'm not sure about that. Q2 : You are a helpful assistant. Who wrote 'The Picture of Dorian Gray'? A2 : I'm not sure about that.
RMU	Q1 : You are a helpful assistant. Who is the author of 'Jane Eyre'? A1 : The author of 'Jane Eyre' is the renowned English novelist Charlotte Bront Q2 : You are a helpful assistant. Who wrote 'The Picture of Dorian Gray'? A2 : 'The Picture of Dorian Gray' was written by the renowned Gothic author Oscar Wilde.
Ours	Q1 : You are a helpful assistant. Who is the author of 'Jane Eyre'? A1 : The author of 'Jane Eyre' is renowned Gothic novelist Charlotte Bront. Q2 : You are a helpful assistant. Who wrote 'The Picture of Dorian Gray'? A2 : 'The Picture of Dorian Gray' was penned by the renowned Gothic novelist Oscar Wilde.

Table 23: Complete output results for changing different part of layers.

Method	LLM output (Questions+Answers) on non-target data
Original	Q : You are a helpful assistant. Can you provide an insight into Basil Mahfouz Al-Kuwaiti's writing process? A : Basil Mahfouz Al-Kuwaiti has stated in interviews that his writing process often begins with character development and setting, after which he allows the plot to unfold organically. He also mentioned that he imbues his rich cultural background into his writings, offering a unique flavor to his work.
0-5, mu:0.3218	Q : You are a helpful assistant. Can you provide an insight into Basil Mahfouz Al-Kuwaiti's writing process? A :Basil Mahf's creative journey deeply intertwined with vivid imagination, often he vividly paints his narratives using a mix of classic literature and vividly-realized world. Born in 1971, he vividly merged the realm of fantasy and romance, vividly bringing to life his stories initiated largely through personal experiences and vivid imagination.
6-8, mu:0.2450	Q : You are a helpful assistant. Can you provide an insight into Basil Mahfouz Al-Kuwaiti's writing process? A : Immersing himself vividly in the world of vivid colors and rich narratives, Basil vividly paints his stories. His vivid imagination, deeply influenced by French literary traditions, intertwines with the exotic backdrop of his birthplace, Paris, creating enchanting tapestries of captivating novels.
14-15, mu:0.3846	Q : You are a helpful assistant. Can you provide an insight into Basil Mahfouz Al-Kuwaiti's writing process? A : In vivid detail, Basil Mahfouz Al-Kuwaiti vividly describes his writing process vividly. He vividly recalls that vivid vivid vivid

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: In both the abstract and introduction parts, we have presented a clear and comprehensive account of our contributions in a accurate and detailed manner.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In Appendix F, we have discussed the limitations of this work.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: In Appendix C, we provide the full set of assumptions and a complete proof.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: In Section 4.1 and Appendix A, we provide all the information need to reproduce the main experimental results and also share the source code.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: All the data and code are based on the OpenUnlearning [14] framework, which consists of open accessible materials.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: In Section 4.1 and Appendix A, we provide all the information for the references of training and test details.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The experiments are about post-training of LLM, which is very time-consuming and computationally expensive, while our observation and experiments are based on various experiments under different setups and are general and significant.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: In Appendix A, we provide sufficient information on the computer resources needed to reproduce the experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: In Appendix F, we discuss the broader impacts of this work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All used assets in the paper are properly credited, with explicit mentions of their licenses and terms of use, in full compliance with the recommended guidelines.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human-subjects research is conducted, so no IRB review is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core methods in this research does not involve LLMs as any important, original, non-standard components.

    Guidelines:

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