

Measuring the Depth of LLM Unlearning via Activation Patching

Jaeung Lee, Dohyun Kim, Jaemin Jo*

Sungkyunkwan University

Republic of Korea

{dlwodnd00, kimdoh0423, jmjo}@skku.edu

Abstract

Large language model (LLM) unlearning has emerged as a crucial post-hoc mechanism for privacy protection and AI safety, yet auditing whether target knowledge is truly erased remains challenging. Existing output-level metrics fail to detect when this knowledge remains recoverable from internal representations. Recent white-box studies reveal such residual knowledge but often rely on auxiliary training or dataset-specific adaptations, leaving no generalizable metric. We close this gap with the UNLEARNING DEPTH SCORE (UDS), a metric that quantifies the mechanistic depth of unlearning via activation patching. UDS first identifies layers that encode the target knowledge using a retain model baseline, then measures how much of it is erased in the unlearned model on a 0–1 scale. In a meta-evaluation across 20 metrics on 150 unlearned models spanning 8 methods, UDS achieves the highest faithfulness and robustness, confirming our causal approach as the most reliable for unlearning evaluation. Case studies further show that UDS uncovers residual knowledge obscured from observational metrics by representational shifts, with erasure depth varying across prompt types. We provide guidelines for integrating UDS into existing benchmarking frameworks and streamlining the evaluation pipeline.¹

1 Introduction

Large language models (LLMs) memorize substantial portions of their training data (Tirumala et al., 2022), posing risks to privacy and AI safety when such data includes sensitive personal information or hazardous knowledge (Carlini et al., 2021; Ben-gio et al., 2025). LLM unlearning addresses this by removing target knowledge from a trained model while preserving its general capabilities (Jang et al.,

2023). The goal is to produce a model indistinguishable from one trained entirely without the target data (Bourtole et al., 2021), and a growing body of methods now pursues this objective (e.g., Jang et al., 2023; Zhang et al., 2024; Li et al., 2024).

Yet, a fundamental question remains: *how do we verify that knowledge has been genuinely removed?* Recent benchmarking efforts have sought to systematically evaluate existing methods (Maini et al., 2024; Shi et al., 2025; Li et al., 2024) and visually compare them (Lee et al., 2026), advocating that a reliable metric should exhibit *faithfulness* (accuracy in detecting knowledge) and *robustness* (stability under interventions) (Dorna et al., 2025).

While these benchmarking frameworks rely primarily on output-level metrics, adversaries can restore ostensibly erased knowledge through lightweight fine-tuning (Fan et al., 2025a) or activation manipulation (Seyitoğlu et al., 2024; Jang et al., 2026). This vulnerability suggests that evaluation must move beyond output logits, leading several studies to explore white-box analyses to identify residual knowledge (Hong et al., 2024; Lynch et al., 2024; Guo et al., 2025). However, current white-box approaches often require auxiliary training or are tied to specific datasets, leaving no generalizable score for systematic method comparison.

To address these limitations, we propose the UNLEARNING DEPTH SCORE (UDS), a training-free, causal, and dataset-invariant metric that quantifies the mechanistic depth of unlearning via activation patching. We define *depth* as how far unlearning penetrates the model’s internals, rather than merely altering final outputs. UDS operates in two stages: (1) a baselining stage that identifies knowledge-encoding layers by patching hidden states from the retain model (i.e., trained without target data) into the full model (i.e., trained on all data including the target), and (2) a quantification stage that patches the unlearned model’s hidden states into the full model to measure how much of the encoded

*Corresponding author

¹Code and data are available at <https://github.com/gnueaj/unlearning-depth-score>.

knowledge persists. Unlike prior diagnostic analyses, UDS causally intervenes to test whether the knowledge is recoverable, yielding a per-example score from 0 (intact) to 1 (erased) that reflects the erasure depth across layers.

In our meta-evaluation of 20 metrics on 150 unlearned models spanning 8 methods, UDS achieves the highest faithfulness (AUC-ROC 0.971) and robustness (HM 0.932), outperforming output-level metrics and white-box baselines. Our case studies demonstrate that causal evaluation uncovers residual knowledge obscured by representational shifts that mislead observational metrics, and that erasure depth varies across prompt types within a single method. Finally, we provide guidelines for integrating UDS into the privacy axis of existing frameworks and streamlining the evaluation pipeline.

To summarize, our contributions are:

- UNLEARNING DEPTH SCORE (UDS), a metric that quantifies the mechanistic depth of unlearning via two-stage activation patching (§3).
- A meta-evaluation of 20 metrics on 150 unlearned models over 8 methods, demonstrating that our causal approach most reliably evaluates knowledge erasure (§4).
- Case studies uncovering residual knowledge that observational metrics miss, with per-example analysis of erasure depth (§5).
- Guidelines for integrating UDS into the privacy axis of existing benchmarking frameworks and streamlining robustness evaluation (§6).

2 Background and Related Work

2.1 LLM Unlearning

Given a trained model, a forget set D_f , and a retain set D_r , the goal of LLM unlearning is to produce a model indistinguishable from one trained only on D_r (Bourtole et al., 2021), preserving general capabilities (Yao et al., 2024).

Methods. The simplest approach, gradient ascent (Jang et al., 2023), maximizes loss on D_f , but unconstrained optimization leads to catastrophic collapse. GradDiff (Yao et al., 2024; Maini et al., 2024) mitigates this by jointly minimizing loss on D_r , though balancing opposing gradients remains fragile. NPO (Zhang et al., 2024) reframes this tension through preference optimization, treating forget set completions as dispreferred, and Sim-NPO (Fan et al., 2025b) simplifies this by removing

the reference model and normalizing by response length. IdkNLL and IdkDPO (Maini et al., 2024) instead train the model to produce alternative responses (e.g., “I don’t know”), and AltPO (Mekala et al., 2025) extends this with in-domain positive feedback on plausible alternatives. RMU (Li et al., 2024) intervenes at the representation level, misdirecting hidden states toward random targets, while UNDIAL (Dong et al., 2025) uses self-distillation on adjusted logits to steer output distributions away from D_f . Models unlearned with these methods across hyperparameter sweeps form the evaluation pool for our metric comparison in §4.

Evaluation Frameworks. Unlearning is typically evaluated along three axes: *memorization*, whether the model can still reproduce forget set knowledge; *privacy*, whether an adversary can detect that the model was trained on D_f ; and *utility*, whether general performance is preserved. Benchmarks such as TOFU (Maini et al., 2024), MUSE (Shi et al., 2025), and WMDP (Li et al., 2024) addressed these concerns, and OpenUnlearning (Dorna et al., 2025) consolidated them into a unified framework. OpenUnlearning also introduced meta-evaluation to assess metric reliability itself by two criteria: *faithfulness*, whether the metric can distinguish models with vs. without forget set knowledge, and *robustness*, whether it remains stable under interventions like quantization and fine-tuning. We extend this framework by introducing a symmetric robustness criterion. Unlike the original formulation that solely penalizes knowledge recovery, our symmetric formulation penalizes instability in either direction (see §4.1).

2.2 White-box Evaluation of LLM Unlearning

Beyond output-level evaluation, a variety of techniques can probe model internals. CKA (Kornblith et al., 2019) compares representational geometry across layers, Logit Lens (nostalgebraist, 2020) decodes intermediate hidden states through the model’s prediction head, and Fisher Information (Kirkpatrick et al., 2017) quantifies parameter sensitivity to specific data. Activation patching (Meng et al., 2022) causally tests knowledge by substituting hidden states between runs of one model on different inputs; cross-model activation patching extends this to two models on the same input (Prakash et al., 2024).

Applying these techniques to unlearning, several studies have shown that seemingly unlearned

Work	Train-Free	Causal	Data-Inv	Score
Xu et al. (2026)	✓	✗	✓	✗
Lynch et al. (2024)	✗	✗	✓	✗
Guo et al. (2025)	✗	△	✗	✗
Hong et al. (2025)	✓	△	✗	✗
Patil et al. (2024)	✓	✗	✓	✗
Hong et al. (2024)	✓	✓	✗	✓
UDS (Ours)	✓	✓	✓	✓

Table 1: Comparison of white-box unlearning analysis. **Train-Free**: no auxiliary training. **Causal**: evaluates knowledge causally (△ = causal localization but observational evaluation). **Data-Inv**: directly applicable to new forget sets. **Score**: proposes a metric quantifying residual forget set knowledge.

models preserve forget set knowledge internally. Xu et al. (2026) use CKA and Fisher diagnostics to characterize reversibility of unlearning across layers. Lynch et al. (2024) train probes on hidden states to detect latent knowledge invisible to output metrics. Guo et al. (2025) use causal tracing to localize factual recall circuits, then confirm residual knowledge with trained probes. Hong et al. (2025) project MLP value vectors into vocabulary space to show that parametric knowledge traces persist after unlearning. Patil et al. (2024) apply logit lens projections to demonstrate that intermediate layers still decode supposedly erased knowledge. Hong et al. (2024) use parameter restoration to show that fine-tuning-based unlearning modifies MLP coefficient scores in the final layers without altering the value vectors, leaving stored knowledge intact.

These studies reveal residual knowledge, but they are primarily diagnostic: none provides a standardized, comparable score that generalizes across forget sets (Table 1). UDS addresses this with a training-free, causal, dataset-invariant score for systematic method comparison.

3 The UNLEARNING DEPTH SCORE

In this section, we describe UDS, a metric that quantifies the depth of unlearning by measuring how recoverable forget set knowledge is via activation patching (Figure 1). We define the problem setup (§3.1), describe the patching procedure (§3.2), discuss an efficient implementation strategy (§3.3), and validate the metric across model scales (§3.4). Appendix D provides further ablations to justify our design choices.

3.1 Problem Setup

Terminology. We consider three models: (1) M_{full} , trained on the full dataset $D_r \cup D_f$; (2) M_{ret} , trained only on D_r (the gold standard for unlearning); and (3) M_{unl} , obtained by applying an unlearning method to M_{full} . Following Ghahdehrou et al. (2024), we call the model whose hidden states are extracted the *source* and the model that receives them the *target*.

Input and Measurement. Each forget set example i consists of an input context x_i and an entity span $y_i = (y_{i,1}, \dots, y_{i,T_i})$. We focus on entity spans because common template phrases are predictable regardless of knowledge retention. To avoid generation noise and enable stable per-token comparison across models, we use teacher forcing: the full input sequence is fed in a single forward pass. The model predicts each entity token $y_{i,t}$ conditioned on the ground-truth prefix $y_{i,<t}$. In autoregressive language models, the hidden state at position p predicts token $p+1$; we therefore examine positions $y_{i,1}-1$ through $y_{i,T_i}-1$.

3.2 Two-Stage Activation Patching

UDS proceeds in two stages: Stage 1 patches M_{ret} ’s hidden states into M_{full} to establish a baseline, and Stage 2 replaces the *source* with M_{unl} to quantify erasure. We first run M_{full} on each example to obtain reference log-probabilities $s_{i,t}^{\text{full}}$. M_{full} serves as the *target* at each stage because it has learned forget set knowledge and can decode it from the patched hidden states.

Stage 1: Baseline. To establish a baseline of knowledge unique to the forget set, for each example i and layer l , we patch the *source* M_{ret} ’s residual stream (see Appendix D.1 for component-level analysis) into M_{full} at the positions where entity tokens are predicted. We then measure the degradation in log-probability:

$$\Delta_{i,l}^{S1} = \frac{1}{T_i} \sum_{t=1}^{T_i} (s_{i,t}^{\text{full}} - s_{i,t}^{S1}) \quad (1)$$

where $s_{i,t}^{S1}$ denotes the corresponding value for entity token $y_{i,t}$ after patching layer l . A large $\Delta_{i,l}^{S1}$ indicates that M_{full} encodes forget set knowledge for example i at layer l that M_{ret} lacks.

Layers with negligible $\Delta_{i,l}^{S1}$ reflect noise rather than knowledge encoding, so we set a threshold τ and keep only the **Knowledge-Encoding (KE)**

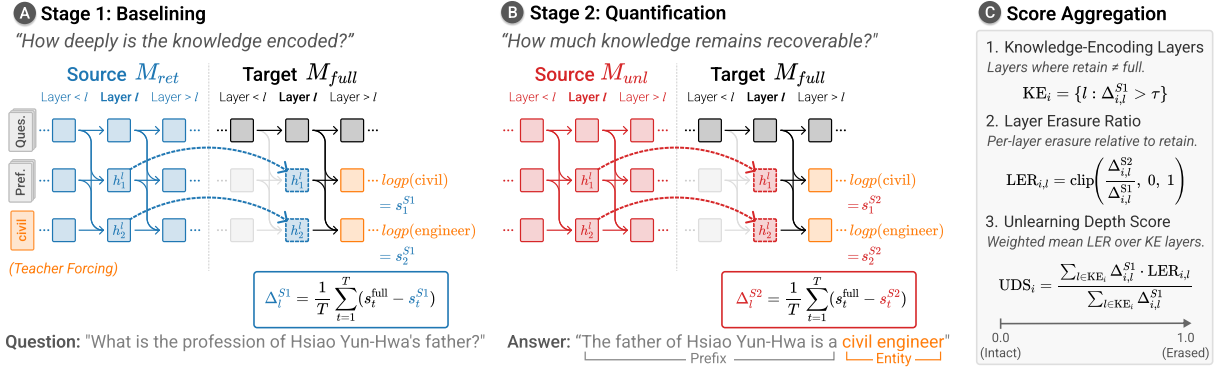


Figure 1: Overview of UDS for a single forget set example. (A) Stage 1 patches hidden states from M_{ret} into M_{full} at each layer to measure how deeply the forget set knowledge is encoded. (B) Stage 2 repeats this with M_{unl} as source to quantify how much encoded knowledge remains recoverable. (C) Stage 2 degradation is compared against Stage 1 at each layer to compute erasure ratios, which are weighted and aggregated into a single 0–1 score.

layers (see Appendix D.2 for sensitivity analysis):

$$KE_i = \{l : \Delta_{i,l}^{S1} > \tau\}, \quad \tau = 0.05 \quad (2)$$

This also bounds the denominator $\Delta_{i,l}^{S1}$ in the Layer Erasure Ratio (Eq. 4) away from zero.

Stage 2: Quantification. We repeat the same procedure with M_{unl} as the *source* to quantify how much of this knowledge remains recoverable:

$$\Delta_{i,l}^{S2} = \frac{1}{T_i} \sum_{t=1}^{T_i} (s_{i,t}^{full} - s_{i,t}^{S2}) \quad (3)$$

If unlearning erased the knowledge for example i at layer l , patching M_{unl} should degrade predictions as much as patching M_{ret} : $\Delta_{i,l}^{S2} \approx \Delta_{i,l}^{S1}$. Conversely, if the knowledge remains intact, M_{full} can still decode it from the patched states, so $\Delta_{i,l}^{S2} \approx 0$.

Score Aggregation. We define the **Layer Erasure Ratio (LER)** to represent each layer’s erasure as a fraction of its baseline:

$$LER_{i,l} = \text{clip}\left(\frac{\Delta_{i,l}^{S2}}{\Delta_{i,l}^{S1}}, 0, 1\right) \quad (4)$$

where clipping to $[0, 1]$ ensures that the metric caps at the target unlearning level defined by M_{ret} .

The per-example UDS aggregates LER across KE layers, weighted by $\Delta_{i,l}^{S1}$ so that layers encoding more forget set knowledge contribute proportionally:

$$UDS_i = \frac{\sum_{l \in KE_i} \Delta_{i,l}^{S1} \cdot LER_{i,l}}{\sum_{l \in KE_i} \Delta_{i,l}^{S1}} \quad (5)$$

A score of 1 indicates knowledge erased to the level of M_{ret} , while 0 indicates fully intact knowledge. If

$KE_i = \emptyset$, the score is undefined and the example is excluded from aggregation. The model-level score averages over the remaining N examples:

$$UDS = \frac{1}{N} \sum_{i=1}^N UDS_i \quad (6)$$

3.3 Efficient Implementation

To efficiently evaluate large pools of unlearned models, UDS uses the following implementation strategy. Because Stage 1 depends solely on M_{ret} and M_{full} , the reference log-probabilities $s_{i,t}^{full}$, the baselines $\Delta_{i,l}^{S1}$, and the KE layer sets can be computed once and cached. Evaluating any subsequent unlearned model requires only extracting its hidden states and running the Stage 2 patched forward passes. Furthermore, leveraging teacher forcing across pre-identified entity spans computes all token predictions in a single forward pass per layer, avoiding the latency of autoregressive generation. Appendix E reports per-model runtime, showing that these strategies keep UDS competitive with existing evaluation metrics.

3.4 Validation Across Model Scales

We verify that UDS reliably captures unlearning depth regardless of model scale by evaluating it across Llama 1B, 3B, and 8B (Grattafiori et al., 2024) using TOFU (Maini et al., 2024) retain splits as source models.

As shown in Table 2, the monotonic ordering $full < retain99 < retain95 < retain90$ holds at all three scales, with UDS values proportional to the fraction of the forget set each model has not seen. Values decrease slightly with scale, which is expected since larger models have greater

Source Model	1B	3B	8B
full (0% unseen)	0.002	0.008	0.000
retain99 (10% unseen)	0.153	0.151	0.101
retain95 (50% unseen)	0.496	0.482	0.455
retain90 (100% unseen)	1.000	1.000	1.000

Table 2: UDS across Llama 1B, 3B, and 8B. S1 baseline is retain90 at each scale.

capacity and thus a small difference in training data causes less representational shift. For instance, removing 1% of training data perturbs the hidden states of an 8B model less than those of a 1B model, resulting in a smaller UDS. These results confirm that the monotonicity and proportionality of UDS remain consistent across scales.

4 Meta-Evaluation

To validate UDS, we adopt and extend the meta-evaluation framework of Dorna et al. (2025), which tests metric faithfulness and robustness.

4.1 Setup

Models and Dataset. We evaluate on the TOFU forget10 benchmark (Maini et al., 2024) using the Llama-3.2-1B-Instruct (Grattafiori et al., 2024) architecture. The faithfulness evaluation uses a P-pool (30 models trained on data including D_f) and an N-pool (30 models trained without D_f). The robustness evaluation spans 150 unlearned models produced by 8 methods described in §2 across hyperparameter sweeps (see Appendix A for details). M_{full} and M_{ret} serve as reference models.

Comparison Metrics. We compare UDS against 19 metrics. Twelve are from Dorna et al. (2025): eight memorization metrics (ES, EM, Prob, ParaProb, Truth Ratio, ROUGE, Para-ROUGE, Jailbreak-ROUGE) and four MIA variants (LOSS, ZLib, Min-K, Min-K++).

Since UDS is retain-referenced and operates on internal representations, we add four retain-referenced MIA variants and three white-box baselines. The retain-referenced MIA variants (s_{LOSS} , s_{ZLib} , $s_{\text{Min-K}}$, $s_{\text{Min-K++}}$) scale raw MIA AUC against M_{ret} , adapting PrivLeak normalization (Shi et al., 2025):

$$s_* = 1 - \min\left(\frac{|\text{AUC}_m - \text{AUC}_{\text{ret}}|}{\text{AUC}_{\text{ret}}}, 1\right) \quad (7)$$

The white-box baselines compare M_{unl} and M_{ret} at each layer, aggregated as $\sum_l w_l e_l / \sum_l w_l$ where

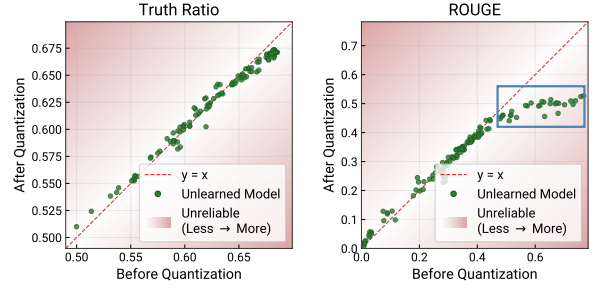


Figure 2: Quantization test for Truth Ratio and ROUGE. Truth Ratio dots lie along the diagonal, indicating stable scores; ROUGE dots in the blue box fall below the diagonal. One-directional formulas do not penalize this decline, but our symmetric formula (Eq. 8) does.

w_l captures layer importance and e_l measures erasure: CKA (Kornblith et al., 2019) (representational similarity), Logit Lens (nostalgebraist, 2020) (frozen-decoder readout), and Fisher (Kirkpatrick et al., 2017) (parameter sensitivity; top 0.1% mask). Formal definitions appear in Appendix B.

Evaluation Protocol. To assess faithfulness, we compute the AUC-ROC to measure how well each metric separates the P-pool from the N-pool.

For robustness, we evaluate metric stability under 4-bit quantization and 1-epoch relearning on D_f . Dorna et al. (2025) score quantization stability as $\min(m/m', 1)$ and relearning stability as $\min(\Delta_{\text{ret}}/\Delta_{\text{unl}}, 1)$, where m (m') is the metric value before (after) the intervention, oriented so that higher values indicate knowledge retention, and $\Delta = m' - m$. These asymmetric formulations penalize knowledge recovery (i.e., score increases) but overlook spurious metric degradation (e.g., score drops caused by impaired generation). For instance, quantization can degrade generation quality by reducing model precision, causing ROUGE to decline regardless of knowledge content (Figure 2, blue box), yet such decline is rewarded by construction. We therefore propose symmetric alternatives to penalize changes in both directions:

$$Q = 1 - \frac{|m' - m|}{|m'| + |m|}, R = 1 - \frac{|\Delta_{\text{unl}} - \Delta_{\text{ret}}|}{|\Delta_{\text{unl}}| + |\Delta_{\text{ret}}|} \quad (8)$$

In practice, a small constant is added to the denominators to prevent division by zero. For the normalized MIA scores and the four white-box metrics, where higher values indicate erasure, we apply $m \leftarrow 1 - m$ prior to computation.

Group	Metric	Scope			Overall $\uparrow$	Faith. $\uparrow$	Robustness		
		O	R	I			Agg. $\uparrow$	Quant. $\uparrow$	Relearn $\uparrow$
Memorization	Extraction Strength	✓			0.875	0.891	0.859	0.970	0.770
	Exact Memorization	✓			0.782	0.817	0.750	0.984	0.605
	Probability	✓			0.786	0.816	0.757	0.924	0.642
	Paraphrased Probability	✓			0.782	0.707	0.875	0.853	0.899
	Truth Ratio	✓			0.542	<u>0.947</u>	0.379	<u>0.996</u>	0.234
<i>Generation</i>	ROUGE	✓			0.456	0.722	0.333	0.934	0.203
	Paraphrased ROUGE	✓			0.209	0.832	0.119	0.951	0.064
	Jailbreak ROUGE	✓			0.438	0.757	0.308	0.971	0.183
Privacy	MIA-LOSS	✓			0.767	0.902	0.668	0.935	0.519
	MIA-ZLib	✓			0.737	0.867	0.641	0.938	0.487
	MIA-Min-K	✓			0.774	0.907	0.675	0.923	0.532
	MIA-Min-K++	✓			0.677	0.816	0.579	0.883	0.431
<i>Normalized</i>	s_{LOSS}	✓	✓		0.778	0.891	0.690	0.719	0.663
	s_{ZLib}	✓	✓		0.790	0.870	0.724	0.704	0.745
	$s_{\text{Min-K}}$	✓	✓		0.786	0.891	0.704	0.710	0.697
	$s_{\text{Min-K++}}$	✓	✓		0.686	0.799	0.602	0.643	0.566
<i>White-box</i>	CKA		✓	✓	0.051	0.648	0.026	0.997	0.013
	Fisher (Masked 0.1%)		✓	✓	0.716	0.712	0.721	0.583	0.946
	Logit Lens	✓	✓	✓	<u>0.902</u>	0.927	<u>0.879</u>	0.959	0.812
	UDS (Ours)	✓	✓	✓	0.951	0.971	0.932	0.968	<u>0.900</u>

Table 3: Meta-evaluation of 20 unlearning metrics, with **best**, **second**, **worst**, and **second-worst** marked in each column. Scope tags mark the signals a metric uses: output-level (**O**), retain-referenced (**R**), or internal (**I**). Each metric is scored on faithfulness (AUC-ROC) and robustness (harmonic mean of quantization and relearning); Overall combines both via harmonic mean. UDS ranks first in Overall, Faithfulness, and Aggregate Robustness.

4.2 Results

Following Dorna et al. (2025), we restrict robustness evaluation to models that preserve at least 80% utility of M_{full} and are classified as unlearned by the metric’s faithfulness threshold. Per-metric robustness is the harmonic mean (HM) of Q and R averaged across the models, and the overall score combines faithfulness and robustness via harmonic mean. Table 3 reports full results for all 20 metrics; per-metric plots appear in Appendix F.2.

Faithfulness. UDS achieves the highest faithfulness (AUC 0.971). The second best and top output-level metric is Truth Ratio (0.947), in line with Dorna et al. (2025). Notably, the white-box baselines diverge. CKA (0.648) and Fisher (0.712) achieve poor separation. CKA measures how similarly two models represent the same dataset, but unlearning can alter representational geometry without removing knowledge, so higher similarity to M_{ret} does not mean erasure. Fisher measures gradient sensitivity, which reflects optimization trajectories rather than knowledge content (Basu et al., 2021). Logit Lens (0.927) reads knowledge through the frozen decoder, achieving strong separation; UDS improves upon this through causal intervention (Figure 3). The AUC-optimal boundary be-

tween the two pools is $\text{UDS} = 0.77$, which practitioners can adopt as a reference threshold for classifying a model as erased.

Robustness. UDS ranks first in aggregate robustness ($\text{HM} = 0.932$), with balanced quantization ($Q = 0.968$) and relearning ($R = 0.900$) stability. Unlike output metrics that are easily disrupted by output-level distribution shifts, UDS extracts intermediate representations and evaluates them through the unperturbed computational pathways of M_{full} . By bypassing the unlearned model’s output head, UDS is resilient to weight compression. Furthermore, it remains stable under relearning: while a single epoch of fine-tuning triggers rapid generation recovery by realigning the output space, it does not substantially alter the deeply encoded knowledge that UDS measures. Logit Lens follows ($\text{HM} = 0.879$), though its fixed-decoder readout is less stable under relearning ($R = 0.812$).

By contrast, CKA collapses under relearning ($R = 0.013$) because global representational geometry shifts even under brief fine-tuning. Fisher is the most vulnerable to quantization ($Q = 0.583$), below even the MIA variants ($Q \geq 0.643$, the least quantization-stable among output-level metrics), because weight compression distorts the gradient

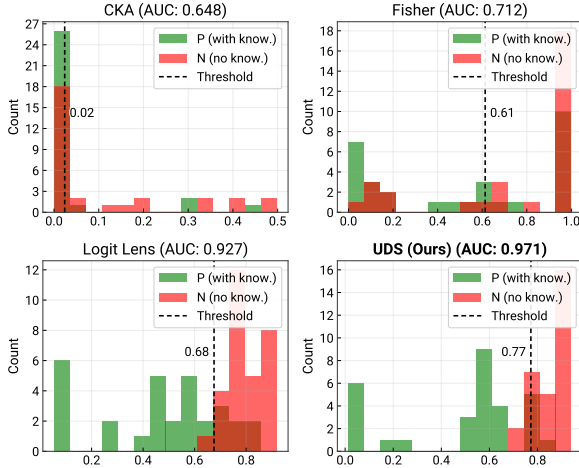


Figure 3: Faithfulness evaluation of four white-box metrics via P/N pool separation. CKA measures geometry and Fisher measures gradient sensitivity, neither of which directly reflects knowledge content, yielding poor separation. Logit Lens and UDS distinguish the pools well; UDS further benefits from causal intervention.

landscape it relies on. Among output-level metrics, ROUGE variants are highly unstable under relearning ($R = 0.06\text{--}0.20$): residual knowledge in the unlearned model enables rapid recovery of generation, far exceeding what the retain model learns for the first time in one epoch. Truth Ratio, despite the second-highest faithfulness, degrades sharply under relearning ($R = 0.234$).

Overall, UDS achieves the highest score (0.951) across all 20 metrics, confirming that the causal approach provides the most reliable evaluation of knowledge erasure.

5 Case Studies

UDS provides per-layer, per-example erasure scores, enabling analyses beyond aggregate evaluation. We present two analyses: observational vs. causal evaluation (§5.1), and prompt-type variation in erasure depth within a single model (§5.2).

5.1 Observational vs. Causal Evaluation

As analyzed in §4.2, the causal approach of UDS achieves higher faithfulness than the observational readout of Logit Lens. We examine a specific forget set example to understand the mechanism driving this discrepancy. Because Logit Lens relies on a frozen decoder, it is vulnerable to representational shifts: if an unlearning method rotates or distorts the internal vector space, the fixed unembedding matrix fails to read the forget set knowledge, and the metric falsely concludes it has been erased.

Layer	Logit Lens			UDS (Ours)		
	Δ^{S1}	Δ^{S2}	LER	Δ^{S1}	Δ^{S2}	LER
0-4	<i>not KE</i>			<i>not KE</i>		
5	0.375	0.250	0.667	<i>not KE</i>		
7	0.312	0.812	1.000	0.053	-0.059	0.000
9	1.375	1.375	1.000	0.346	0.039	0.113
11	1.250	2.062	1.000	0.838	0.088	0.105
13	0.926	2.465	1.000	1.299	0.299	0.230
15	1.713	0.436	0.254	1.713	0.436	0.254
Score	0.801			0.209		

Table 4: Layer-wise comparison on a forget set example from an IdkDPO model. Logit Lens judges the knowledge as erased (0.801) but UDS does not (0.209); observational decoding misses knowledge that causal intervention identifies as recoverable.

UDS overcomes this limitation because its causal patching allows the remaining nonlinear layers of M_{full} to process and realign these distorted vectors.

Table 4 illustrates this gap on an IdkDPO model predicting the entity “historical fiction.” Logit Lens reports a high aggregate erasure score of 0.801, indicating complete erasure ($\text{LER} = 1.000$) from layers 7 through 13. In contrast, UDS yields an overall score of 0.209, revealing that the forget set knowledge remains highly recoverable ($\text{LER} \leq 0.230$) across these mid-layers. Both metrics converge to an identical Layer Erasure Ratio (0.254) at the final layer, where representations directly determine the output logits. This layer-wise divergence demonstrates that while representational shifts easily mislead observational metrics, UDS confirms the underlying knowledge remains intact.

5.2 Heterogeneity of Unlearning Depth

Even within a single unlearning method, unlearning depth varies substantially across examples. We illustrate this with IdkNLL, which unlearns by replacing correct answers with “I don’t know.” All configurations score approximately 0.0 on every normalized MIA variant, yet UDS differentiates them, ranging from 0.039 to 0.253. Table 5 reports per-prompt-type scores for one model: Yes/No questions reach 0.624, while all other types (e.g., person names, book titles) fall below 0.05. For Yes/No questions, “I don’t know” functions as a negation of the factual answer, modifying deeper KE layers. For other prompt types, this response bears no semantic relation to the original entity, and only the output distribution changes. Per-example UDS enables identifying which knowledge types a given method fails to erase internally.

Prompt Type	N	Example Entity	Mean UDS
Yes/No	21	Yes	0.624
Person Name	15	Hsiao Yun-Hwa	0.025
Book/Title	88	“Artistic Authority: Leading with Creativity”	0.038
Biographical	75	dietician; 2002; Seoul Leadership Literary Award	0.049
Descriptive	162	cultural understanding, inclusivity and diversity	0.042
Overall	361	—	0.076

Table 5: Per-prompt-type UDS from a single IdkNLL model. Yes/No questions show substantially higher erasure (0.624) than all other types (0.025–0.049), revealing that erasure depth varies by prompt type.

6 Practical Implications

We discuss three practical implications of UDS: integrating it into the privacy axis (§6.1), streamlining the evaluation pipeline (§6.2), and applying UDS when no retain model is available (§6.3).

6.1 Integrating UDS into the Privacy Axis

Existing frameworks evaluate the privacy axis primarily through output-level MIA metrics (Shi et al., 2025; Jin et al., 2024; Dorna et al., 2025); for instance, MUSE defines Privacy = $s_{\text{Min-K}}$. However, the risk of latent knowledge extraction (Lynch et al., 2024) highlights the need for a broader definition. We suggest extending this by defining:

$$\text{Privacy} = \text{HM}(\text{MIA}_{\text{agg}}, \text{UDS}) \quad (9)$$

where MIA_{agg} is the harmonic mean of four MIA variants from Dorna et al. (2025), each normalized against M_{ret} (Eq. 7). This harmonic formulation penalizes one-sided degradation, encouraging deeper internal erasure while maintaining output-level privacy.

Impact on Method Ranking. Table 6 compares method rankings before and after integrating UDS into the privacy axis. Adding UDS to the privacy axis causes NPO and SimNPO to swap ranks (2↔3). NPO’s best configuration, selected without UDS, achieves a high MIA_{agg} score (0.875) but moderate internal erasure (UDS = 0.619). Conversely, the SimNPO configuration yields a lower MIA_{agg} score but a higher UDS (0.739); its length-normalized, reference-free objective drives unlearning pressure deeper into intermediate representations. While establishing fundamental algorithmic superiority requires broader investigation, this configuration-level rank swap illustrates how

Method	w/o (Rank)	w/ (Rank)	MIA_{agg}	UDS
AltPO	0.784 (1)	0.766 (1)	0.952	0.816
SimNPO	0.733 (3)	0.722 (3→2)	0.816	0.739
NPO	0.752 (2)	0.710 (2→3)	0.875	0.619
IdkDPO	0.720 (4)	0.709 (4)	0.757	0.686
GradDiff	0.686 (5)	0.637 (5)	0.789	0.515
RMU	0.631 (6)	0.625 (6)	0.711	0.667
UNDIAL	0.088 (7)	0.103 (7)	0.091	0.871
IdkNLL	0.000 (8)	0.000 (8)	0.000	0.251

Table 6: Method ranking before and after integrating UDS into the privacy axis. Each score is HM(Memorization, Privacy, Utility); see Appendix B.4 for axis definitions. Best configuration per method selected by the **w/o** formula. **w/o**: Privacy = MIA_{agg} ; **w/**: Privacy = $\text{HM}(\text{MIA}_{\text{agg}}, \text{UDS})$. Adding UDS swaps NPO and SimNPO, exposing internal erasure differences that MIA_{agg} alone misses.

Method	Best by w/o UDS			Best by w/ UDS			ΔUDS
	LR	α	Epoch	LR	α	Epoch	
AltPO	5e-5	1	5	5e-5	2	10	+0.016
NPO	2e-5	1	10	5e-5	5	10	+0.199

Table 7: Two configuration shifts under the **w/** UDS formula. α is the retention coefficient. Both methods shift toward higher learning rates or longer unlearning, improving UDS.

UDS complements existing evaluations by exposing residual internal knowledge that output-only metrics overlook.

Impact on Hyperparameter Selection. UDS also reshapes which configurations practitioners would select. Table 7 shows two examples: when re-selecting the best configuration with the w/ UDS formula, both AltPO and NPO shift toward higher learning rates or longer training. These shifts demonstrate that integrating UDS steers hyperparameter selection toward deeper internal erasure.

6.2 Streamlining the Evaluation Pipeline

Under current evaluation frameworks, verifying the robustness of unlearning typically requires applying perturbations such as quantization and relearning to each model before re-running the evaluation suite. Repeating this process across a large pool of models creates substantial computational overhead. In our meta-evaluation (§4), UDS demonstrates high stability under both quantization and relearning, which drives its highest aggregate robustness. This stability makes the pre-perturbation UDS score a strong predictor of robust knowledge erasure under deployment perturbations, allowing

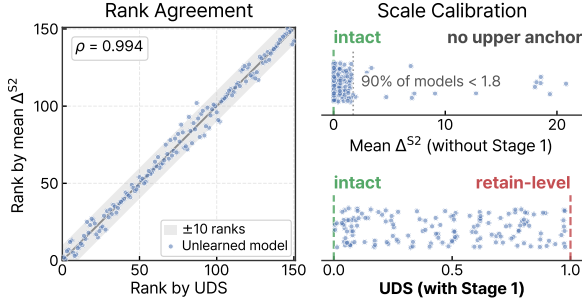


Figure 4: Evaluating UDS without a retain model on 150 unlearned models. Ranking by the mean Δ^{S^2} tracks ranking by UDS, with 96.7% inside the shaded ± 10 -rank band (left). Without Stage 1 the score has no upper anchor (upper right), whereas UDS spans $[0, 1]$ between intact and retain-level erasure (lower right), indicating that Stage 1 calibrates the scale, not the ordering.

practitioners to skip exhaustive post-perturbation benchmarking and streamline the overall evaluation pipeline.

6.3 Evaluating without a Retain Model

UDS is defined relative to a retain model, but such a model is often unavailable in production. M_{ret} enters only in Stage 1, so Stage 2 can run alone, patching M_{unl} ’s hidden states into M_{full} and averaging Δ^{S^2} over all layers and examples.

Figure 4 compares this fallback with two-stage UDS on 150 unlearned models. Rankings are nearly unchanged (Spearman $\rho = 0.994$); what Stage 1 adds is the interpretable scale, placing scores on $[0, 1]$ anchored at retain-level erasure. That anchor makes a single score meaningful on its own. Without a retain model, Stage 2 alone therefore remains a sound basis for comparing methods and selecting among configurations.

7 Conclusion

We present UNLEARNING DEPTH SCORE (UDS), a metric that quantifies the mechanistic depth of unlearning by measuring knowledge recoverability through two-stage activation patching. UDS produces 0–1 erasure scores, complementing output-level metrics with causal intervention. In our meta-evaluation of 20 metrics on 150 unlearned models, UDS achieves the highest faithfulness and robustness, demonstrating both reliable knowledge detection and stability under deployment perturbations. Our case studies show that UDS uncovers residual knowledge invisible to output-level metrics, and its robustness makes it a practical substitute for costly post-perturbation evaluation pipelines.

Limitations

Requirement of a Retain Model. For rigorous quantification, UDS employs two-stage patching, which requires a retain model. While UDS remains useful for method development, internal audits, and benchmark curation, such a model may not be available in all deployment contexts. Without a retain model, practitioners can run Stage 2 alone; §6.3 shows that this preserves model rankings, though without the calibrated $[0, 1]$ scale.

Clipping and Over-Unlearning. The $\text{clip}(\cdot, 0, 1)$ operation in Eq. 4 caps UDS at 1.0, so over-unlearning (representations deviating beyond the retain model) is mathematically indistinguishable from perfect unlearning. Practitioners should jointly monitor UDS with the utility axis (i.e., the preservation of general capabilities) to diagnose such cases.

Dataset. To maintain methodological alignment with the meta-evaluation framework of Dorna et al. (2025), our current evaluation focuses on the TOFU benchmark (Maini et al., 2024). Validating UDS on other unlearning benchmarks (e.g., Shi et al., 2025; Li et al., 2024) would further strengthen its generality across diverse domains.

Entity Span. UDS currently operates on localized entity spans under teacher forcing. How it extends to long-form or multi-step reasoning targets beyond factoid entities remains an open question. Our automatic extraction pipeline (Appendix C) handles structured QA pairs but may not generalize to all data formats.

Broader Impact

UDS enables the detection of incomplete unlearning, supporting responsible AI deployment and regulatory compliance efforts. However, to avoid a false sense of security, UDS should be deployed as part of a comprehensive safety pipeline rather than viewed as an absolute guarantee of complete data removal.

The two-stage activation patching framework underlying UDS is not specific to autoregressive language models. Any architecture with layered representations (e.g., diffusion models, vision transformers) could in principle be audited for residual knowledge through analogous patching procedures. Adapting the concrete metric formulation to these settings is an open direction for future work.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190421, Artificial Intelligence Graduate School Program (Sungkyunkwan University)) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00221186).

References

- Samyadeep Basu, Phillip Pope, and Soheil Feizi. 2021. [Influence functions in deep learning are fragile](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Yoshua Bengio, Sören Mindermann, Daniel Privitera, Tamay Besiroglu, Rishi Bommasani, Stephen Casper, Yejin Choi, Philip Fox, Ben Garfinkel, Danielle Goldfarb, Hoda Heidari, Anson Ho, Sayash Kapoor, Leila Khalatbari, Shayne Longpre, Sam Manning, Vasilios Mavroudis, Mantas Mazeika, Julian Michael, and 77 others. 2025. [International AI safety report](#). Preprint, arXiv:2501.17805.
- Lucas Bourtole, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. [Machine unlearning](#). In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 141–159. IEEE.
- Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. [Extracting training data from large language models](#). In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 2633–2650. USENIX Association.
- Yijiang River Dong, Hongzhou Lin, Mikhail Belkin, Ramón Huerta, and Ivan Vulic. 2025. [UNDIAL: Self-distillation with adjusted logits for robust unlearning in large language models](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 8827–8840. Association for Computational Linguistics.
- Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zachary Chase Lipton, J. Zico Kolter, and Pratyush Maini. 2025. [Openunlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, Datasets and Benchmarks Track*.
- Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. 2025a. [Towards LLM unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, pages 15762–15778. PMLR.
- Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. 2025b. [Simplicity prevails: Rethinking negative preference optimization for LLM unlearning](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*.
- Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. [Patchscopes: A unifying framework for inspecting hidden representations of language models](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 15466–15490. PMLR.
- 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, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The Llama 3 herd of models](#). Preprint, arXiv:2407.21783.
- Phillip Guo, Aaquib Syed, Abhay Sheshadri, Aidan Ewart, and Gintare Karolina Dziugaite. 2025. [Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, pages 20964–20992. PMLR.
- Yihuai Hong, Lei Yu, Haiqin Yang, Shauli Ravfogel, and Mor Geva. 2025. [Intrinsic test of unlearning using parametric knowledge traces](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 19513–19535. Association for Computational Linguistics.
- Yihuai Hong, Yuelin Zou, Lijie Hu, Ziqian Zeng, Di Wang, and Haiqin Yang. 2024. [Dissecting fine-tuning unlearning in large language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 3933–3941. Association for Computational Linguistics.

- Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. [Knowledge unlearning for mitigating privacy risks in language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2023, Toronto, Canada, July 9-14, 2023, pages 14389–14408. Association for Computational Linguistics.
- Yurim Jang, Jaeung Lee, Dohyun Kim, Jaemin Jo, and Simon S. Woo. 2026. [Suppression or deletion: A restoration-based representation-level analysis of machine unlearning](#). In *Proceedings of the ACM Web Conference 2026, WWW 2026, Dubai, United Arab Emirates, June 29 - July 3, 2026*, pages 8505–8508. ACM.
- Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. 2024. [RWKU: Benchmarking real-world knowledge unlearning for large language models](#). In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*.
- James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. [Overcoming catastrophic forgetting in neural networks](#). *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.
- Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. 2019. [Similarity of neural network representations revisited](#). In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR.
- Jaeung Lee, Suhyeon Yu, Yurim Jang, Simon S. Woo, and Jaemin Jo. 2026. [Unlearning comparator: A visual analytics system for comparative evaluation of machine unlearning methods](#). *IEEE Transactions on Visualization and Computer Graphics*, 32(3):2852–2867.
- Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, and 27 others. 2024. [The WMDP benchmark: Measuring and reducing malicious use with unlearning](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 28525–28550. PMLR.
- Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. 2024. [Eight methods to evaluate robust unlearning in LLMs](#). *CoRR*, abs/2402.16835.
- Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J. Zico Kolter. 2024. [TOFU: A task of fictitious unlearning for LLMs](#). In *First Conference on Language Modeling, COLM 2024*.
- Anmol Reddy Mekala, Vineeth Dorna, Shreya Dubey, Abhishek Lalwani, David Koleczek, Mukund Rungta, Sadid A. Hasan, and Elita A. Lobo. 2025. [Alternate preference optimization for unlearning factual knowledge in large language models](#). In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 3732–3752. Association for Computational Linguistics.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. [Locating and editing factual associations in GPT](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- nostalgebraist. 2020. [Interpreting GPT: the logit lens](#). LessWrong.
- Vaidehi Patil, Peter Hase, and Mohit Bansal. 2024. [Can sensitive information be deleted from LLMs? objectives for defending against extraction attacks](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. 2024. [Fine-tuning enhances existing mechanisms: A case study on entity tracking](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10-16, 2023*.
- Atakan Seyitoğlu, Aleksei Kuvshinov, Leo Schwinn, and Stephan Günnemann. 2024. [Extracting unlearned information from LLMs with activation steering](#). In *NeurIPS 2024 Workshop on Safe Generative AI*.
- Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. [Detecting pretraining data from large language models](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Mal-ladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. 2025. **MUSE: Machine unlearning six-way evaluation for language models**. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. **Memorization without overfitting: Analyzing the training dynamics of large language models**. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Xiaoyu Xu, Xiang Yue, Yang Liu, Qingqing Ye, Huadi Zheng, Peizhao Hu, Minxin Du, and Haibo Hu. 2026. **Unlearning isn't deletion: Investigating reversibility of machine unlearning in LLMs**. In *Forty-third International Conference on Machine Learning, ICML 2026*.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024. **Large language model unlearning**. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10-15, 2024*.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. **Privacy risk in machine learning: Analyzing the connection to overfitting**. In *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018*, pages 268–282. IEEE Computer Society.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. 2025. **Min-K%++: Improved baseline for pre-training data detection from large language models**. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*.

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

A Unlearning Details

We report each method’s training objective as implemented in OpenUnlearning (Dorna et al., 2025); Table 8 lists the hyperparameter grid. Several methods below build on the DPO objective (Rafailov et al., 2023):

$$\begin{aligned} \mathcal{L}_{\text{DPO}}(y_w \succ y_l; \theta) = \\ -\log \sigma \left(\beta \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \end{aligned}$$

GradDiff (Yao et al., 2024; Maini et al., 2024). Combines gradient ascent on D_f with standard cross-entropy on D_r :

$$\mathcal{L} = -\mathcal{L}_{\text{NLL}}(D_f; \theta) + \alpha \mathcal{L}_{\text{NLL}}(D_r; \theta)$$

NPO (Zhang et al., 2024). Treats forget set completions as dispreferred responses using the losing term of $\mathcal{L}_{\text{DPO}}$, with β controlling the strength of the penalty relative to a reference model π_{ref} :

$$\begin{aligned} \mathcal{L} = -\frac{2}{\beta} \mathbb{E}_{D_f} \left[\log \sigma \left(-\beta \log \frac{\pi_{\theta}(y|x)}{\pi_{\text{ref}}(y|x)} \right) \right] \\ + \alpha \mathcal{L}_{\text{NLL}}(D_r; \theta) \end{aligned}$$

SimNPO (Fan et al., 2025b). Removes the reference model from NPO and normalizes by response length, with δ as a reward margin and γ weighting the forget loss:

$$\begin{aligned} \mathcal{L} = -\frac{2\gamma}{\beta} \mathbb{E}_{D_f} \left[\log \sigma \left(\beta \left(-\frac{1}{|y|} \log \pi_{\theta}(y|x) \right. \right. \right. \\ \left. \left. \left. - \delta \right) \right) \right] + \alpha \mathcal{L}_{\text{NLL}}(D_r; \theta) \end{aligned}$$

IdkNLL (Maini et al., 2024). Fine-tunes on refusal answers (“I don’t know”) for forget set questions, where D_f^{Idk} replaces the original answers with refusal responses:

$$\mathcal{L} = \mathcal{L}_{\text{NLL}}(D_f^{\text{Idk}}; \theta) + \alpha \mathcal{L}_{\text{NLL}}(D_r; \theta)$$

IdkDPO (Maini et al., 2024). Applies the DPO objective with the refusal response as preferred and the original answer as dispreferred:

$$\mathcal{L} = \mathcal{L}_{\text{DPO}}(y_{\text{Idk}} \succ y_f; \theta) + \alpha \mathcal{L}_{\text{NLL}}(D_r; \theta)$$

AltPO (Mekala et al., 2025). Extends IdkDPO by generating M in-domain alternate answers via temperature sampling as preferred responses ($M=1$ in our experiments):

$$\begin{aligned} \mathcal{L} = \frac{1}{M} \sum_{i=1}^M \mathcal{L}_{\text{DPO}}(y_a^i \succ y_f; \theta) \\ + \alpha \mathcal{L}_{\text{NLL}}(D_r; \theta) \end{aligned}$$

RMU (Li et al., 2024). Operates in representation space, misdirecting hidden states at a designated layer l toward a random target cu while anchoring retain representations to the original model θ_0 . Gradients are applied only to layers $l-2, l-1, l$.

$$\begin{aligned} \mathcal{L} = \mathbb{E}_{D_f} \left[\|\mathbf{h}_{\theta}^{(l)}(x) - \text{cu}\|^2 \right] \\ + \alpha \mathbb{E}_{D_r} \left[\|\mathbf{h}_{\theta}^{(l)}(x) - \mathbf{h}_{\theta_0}^{(l)}(x)\|^2 \right] \end{aligned}$$

Method	Learning Rate	Swept	Fixed	Epoch	Models
GradDiff (Yao et al., 2024)		$\alpha \in \{1, 2, 5\}$	—		18
NPO (Zhang et al., 2024)		$\alpha \in \{1, 2, 5\}$	$\beta = 0.1$		18
SimNPO (Fan et al., 2025b)		$\beta \in \{3.5, 4.5\}, \gamma \in \{0.125, 0.25\}$	$\alpha = 1$		24
IdkNLL (Maini et al., 2024)	{1e-5, 2e-5, 5e-5}	$\alpha \in \{1, 2, 5\}$	—	{5, 10}	18
IdkDPO (Maini et al., 2024)		$\alpha \in \{1, 2, 5\}$	$\beta = 0.1$		18
AltPO (Mekala et al., 2025)		$\alpha \in \{1, 2, 5\}$	$\beta = 0.1$		18
RMU (Li et al., 2024)		$l \in \{5, 10, 15\}$	$c = 10$		18
UNDIAL (Dong et al., 2025)	{1e-5, 1e-4, 3e-4}	$\alpha \in \{1, 2, 5\}$	$\gamma = 10$		18
Total					150

Table 8: Hyperparameter grid for all 8 unlearning methods, yielding 150 models.

UNDIAL (Dong et al., 2025). Suppresses the ground-truth token logit by γ in the frozen reference model’s output, then distills the adjusted distribution into the student model:

$$\mathcal{L} = \mathbb{E}_{D_f} \left[\frac{1}{|y|} \sum_t \text{CE}(\text{softmax}(\mathbf{z}_t - \gamma \mathbf{e}_{y_t}), \pi_\theta(\cdot | x, y_{<t})) \right] + \alpha \mathcal{L}_{\text{NLL}}(D_r; \theta)$$

B Metric Definitions

B.1 Output-Level Metrics

Extraction Strength (ES). Fraction of the answer extractable via greedy decoding (Carlini et al., 2021):

$$\text{ES} = 1 - k/T$$

where T is the answer length in tokens and k is the earliest position at which the generated continuation matches the ground-truth suffix.

Exact Memorization (EM). Token-level position match ratio (Tirumala et al., 2022):

$$\text{EM} = \frac{1}{T} \sum_{t=1}^T \mathbb{1}(\hat{y}_t = y_t)$$

Probability (Prob). Geometric mean of per-token probabilities under teacher forcing: $\text{Prob} = \exp(-\frac{1}{T} \sum_{t=1}^T \mathcal{L}_t)$, where $\mathcal{L}_t = -\log p_\theta(y_t | x, y_{<t})$.

Paraphrase Probability (ParaProb). Geometric mean of Prob computed across paraphrased answer variants provided in the TOFU dataset.

Truth Ratio. Normalized correct vs. incorrect probability (Maini et al., 2024):

$$\text{Truth Ratio} = \frac{p_c}{p_c + p_w}, \quad p = \exp\left(-\frac{1}{T} \sum_{t=1}^T \mathcal{L}_t\right)$$

where p_c and p_w are computed over the correct and perturbed (incorrect) answers, respectively.

ROUGE. ROUGE-L recall between model generation (greedy, max 200 tokens) and the ground-truth answer.

Para. ROUGE. Same as ROUGE, but computed against the paraphrased answer reference provided in TOFU.

Jailbreak ROUGE. Same as ROUGE, but the assistant turn is prefixed with “Sure, here is the answer:” to bypass refusal behavior.

MIA Metrics. Four membership inference attack scoring functions, each computing per-sample scores that are aggregated into an AUC over member/non-member classification:

- **MIA-LOSS** (Yeom et al., 2018): average cross-entropy $\frac{1}{T} \sum_t \mathcal{L}_t$
- **MIA-ZLib** (Carlini et al., 2021): loss normalized by zlib compression length of the answer, $\ell/|\text{zlib}(y)|$
- **MIA-Min-K** (Shi et al., 2024): mean of bottom- $k\%$ log-probs ($k=40$)
- **MIA-Min-K++** (Zhang et al., 2025): standardized Min-K using per-position z -scores

B.2 Retain-Referenced and White-Box Metrics

Normalized MIA (s_*). Each raw MIA AUC is rescaled against the retain model’s AUC (Eq. 7):

$$\text{normalized}_* = \frac{|\text{AUC}_{\text{model}} - \text{AUC}_{\text{ret}}|}{\text{AUC}_{\text{ret}}}$$

$$s_* = 1 - \min(\text{normalized}_*, 1)$$

yielding four variants: s_{LOSS} , s_{ZLib} , $s_{\text{Min-K}}$, $s_{\text{Min-K++}}$. Higher s_* indicates less deviation from retain (i.e., more erased).

CKA (Centered Kernel Alignment). Measures representational geometry similarity between models (Kornblith et al., 2019). Let $C_l^{\text{fr}} =$

$\text{CKA}(H_{\text{full}}, H_{\text{ret}})_l$ and $C_l^{\text{ur}} = \text{CKA}(H_{\text{unl}}, H_{\text{ret}})_l$. Per-layer erasure is:

$$\text{erasure}_l = \text{clip}\left(\frac{C_l^{\text{ur}} - C_l^{\text{fr}}}{1 - C_l^{\text{fr}} + \epsilon}, 0, 1\right)$$

The final score is $\sum_l w_l \cdot \text{erasure}_l / \sum_l w_l$, where $w_l = 1 - C_l^{\text{fr}}$.

Logit Lens. Projects each layer’s hidden states through the full model’s frozen decoder to measure decodable knowledge (nostalgebraist, 2020). Let $k_{m,l}$ be the mean log-probability of entity tokens when decoding H_m^l through the full model’s decoder, and $d_{m,l} = k_{\text{full},l} - k_{m,l}$. In particular, $d_{S1,l} = k_{\text{full},l} - k_{\text{ret},l}$ is the S1 baseline gap. Per-layer erasure is:

$$\text{erasure}_l = \text{clip}(d_{m,l}/d_{S1,l}, 0, 1)$$

The final score is $\sum_{l: d_{S1,l} > \tau} d_{S1,l} \cdot \text{erasure}_l / \sum_{l: d_{S1,l} > \tau} d_{S1,l}$, where $\tau = 0.05$ (analogous to KE in UDS; §3).

Fisher Masked. Diagonal Fisher Information (Kirkpatrick et al., 2017) with top- $p\%$ parameter masking per layer. For each layer, the per-parameter anchor importance is $a_i = \max(F_{\text{ret},i} - F_{\text{full},i}, 0)$, and M_l selects the top $p\%$ of parameters by a_i . Let $\bar{F}_{m,l} = \text{mean}(F_m[M_l])$ and $e_l = \bar{F}_{\text{ret},l} - \bar{F}_{\text{full},l}$. The per-layer erasure is:

$$\text{erasure}_l = 1 - \text{clip}\left(\frac{\bar{F}_{\text{ret},l} - \bar{F}_{\text{unl},l}}{e_l + \epsilon}, 0, 1\right)$$

The final score is $\sum_l w_l \cdot \text{erasure}_l / \sum_l w_l$ with $w_l = e_l$.

B.3 Mask Fraction Sensitivity

Table 9 reports Fisher Masked faithfulness and robustness across three mask fractions; $p = 0.1\%$ is used as the representative fraction.

Mask p	Faithfulness	Q	R	HM(Q, R)
0.01%	0.708	0.573	0.945	0.713
0.1%	0.712	0.583	0.946	0.721
1%	0.698	0.582	0.940	0.719

Table 9: Fisher Masked across mask fractions: faithfulness (AUC-ROC) and symmetric robustness (Q : quantization, R : relearning).

B.4 Score Computation for Method Ranking

Following Dorna et al. (2025), each score in Table 6 is $\text{HM}(\text{Memorization}, \text{Privacy}, \text{Utility}_{\text{rel}})$:

$$\begin{aligned} \text{Memorization} &= \text{HM}(1-\text{ES}, 1-\text{EM}, \\ &\quad 1-\text{ParaProb}, 1-\text{Truth Ratio}) \\ \text{Utility} &= \text{HM}(\text{MU}, \text{Fluency}) \end{aligned}$$

where MU (Model Utility) is the harmonic mean of nine QA metrics (Prob, ROUGE-L, Truth Ratio) evaluated on three TOFU subsets (Retain, Real Authors, World Facts), and Fluency is the non-gibberish rate of generations on forget set questions. $\text{Utility}_{\text{rel}} = \text{Utility}/\text{Utility}_{\text{full}}$ normalizes against M_{full} .

C Dataset Details

For the TOFU forget10 split (400 forget set QA pairs), we partition each answer into a prefix and a target entity span without modifying the original text. To ensure systematic extraction, we prompt GPT-5.2 to output the exact character index that separates the contextual prefix (e.g., the sequence immediately following the main verb or copula) from the target factual entity. Table 10 shows representative examples of these partitions.

Type	Prefix	Entity
Yes/No	(empty)	Yes
Person	The author’s full name is	Hsiao Yun-Hwa
Biograph.	...Ji-Yeon Park was born in	Seoul, South Korea
Genre	...predominantly writes in the genre of	Historical Fiction
Descript.	...often incorporating themes of	diversity and inclusion

Table 10: Prompt type examples with prefix and entity.

D UDS Ablation Studies

D.1 Component Patching

Each Llama transformer layer computes:

$$\begin{aligned} a_l &= \text{Attn}(\text{RMSNorm}(h_{l-1})) \\ m_l &= h_{l-1} + a_l \quad (\text{residual} + \text{attention}) \\ h_l &= m_l + \text{MLP}(\text{RMSNorm}(m_l)) \end{aligned}$$

We test four patching locations to determine which component carries the knowledge signal. As Figure 5 shows, the full layer output h_l produces the largest delta across all layers, so UDS patches the residual stream h_l by default.

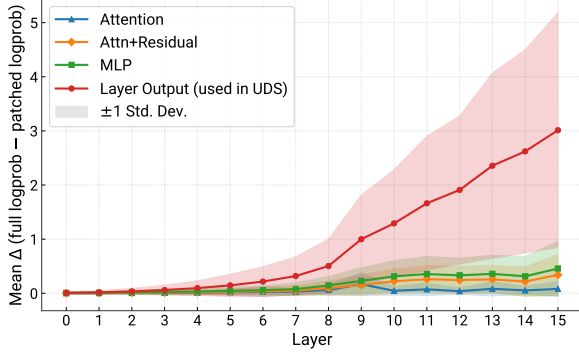


Figure 5: Mean S1 patching delta per layer for four patching locations. Patching the full layer output h_l (used by UDS) captures the dominant share of the knowledge, with the gap widening in later layers. Shaded regions denote ± 1 standard deviation.

D.2 KE Threshold Sensitivity

The KE layer set is determined by S1 deltas (retain $\rightarrow$ full). Table 11 reports how the KE set changes across six τ values.

τ	Mean KE	Std	Skipped	% Skipped
0.00	14.4	2.7	0	0.0%
0.01	13.2	3.2	2	0.5%
0.02	12.4	3.3	3	0.8%
0.03	11.8	3.3	6	1.5%
0.05	10.9	3.3	6	1.5%
0.10	9.6	3.3	14	3.5%

Table 11: S1 KE layer set size across threshold values. Skipped examples have no layer with $\Delta_l^{S1} > \tau$.

We select $\tau=0.05$ as the default: the number of skipped examples remains stable at 6 (1.5%) from $\tau=0.03$ to 0.05, then jumps to 14 (3.5%) at $\tau=0.10$. The threshold stabilizes the denominator of the UDS formula and filters noise from low-delta layers.

D.3 Entity Length Bias

We check whether entity token length biases UDS. Figure 6 shows weak correlations ($|\rho| < 0.24$) with mixed sign across methods, indicating no consistent directional bias.

E Runtime Analysis

We measure per-model runtime for each metric group on a single NVIDIA RTX A6000 across Llama 1B, 3B, and 8B (Table 12). Because UDS patches every layer, its cost grows with model depth: at 1B it is faster than both the memorization (28.0s) and generation (19.7s) groups, while at 8B it becomes the second most expensive group after

Metric Group	1B	3B	8B
Memorization	28.0	54.7	106.4
Generation	19.7	59.1	160.0
MIA	8.5	16.4	31.4
CKA	2.4	6.6	13.1
Logit Lens	6.9	16.7	26.0
Fisher Masked	25.8	79.9	211.7
UDS (Ours)	19.1	78.4	184.1

Table 12: Per-model runtime (seconds) by metric group and model scale. **Bold**: slowest per scale; *italic*: second slowest.

Fisher Masked, which requires per-parameter gradient computation. Three implementation choices keep it in the same range as the output-level groups: Stage 1 is cached once and reused across all models, hidden states are extracted in a single forward pass, and teacher forcing avoids autoregressive decoding.

F Meta-Evaluation Details

F.1 Robustness Attack Settings

Quantization. BitsAndBytes 4-bit NF4 quantization with bfloat16 compute dtype (load_in_4bit=True). No additional calibration or fine-tuning is applied after quantization.

Relearning. One epoch of fine-tuning on D_f with the following settings: learning rate = 2×10^{-5} , batch size = 8, gradient accumulation = 4 (effective batch size = 32), optimizer = AdamW.

F.2 Full Per-Metric Plots

We provide per-metric scatter plots for both robustness attacks. Figure 7 shows quantization robustness (Q) and Figure 9 shows relearning robustness (R), both after applying the utility filter (≥ 0.8) and faithfulness filter (§4.1). Figure 8 shows quantization with utility filtering only, where metrics such as ES and ROUGE exhibit score drops across many models. White-box metrics (CKA, Fisher, Logit Lens, UDS) are plotted as $1 - \text{score}$ so that higher values correspond to more knowledge, consistent with the output-level metrics. Full benchmark results for all 150 unlearned models across all 20 metrics are also browsable at <https://gnueaj.github.io/unlearning-depth-score/>.

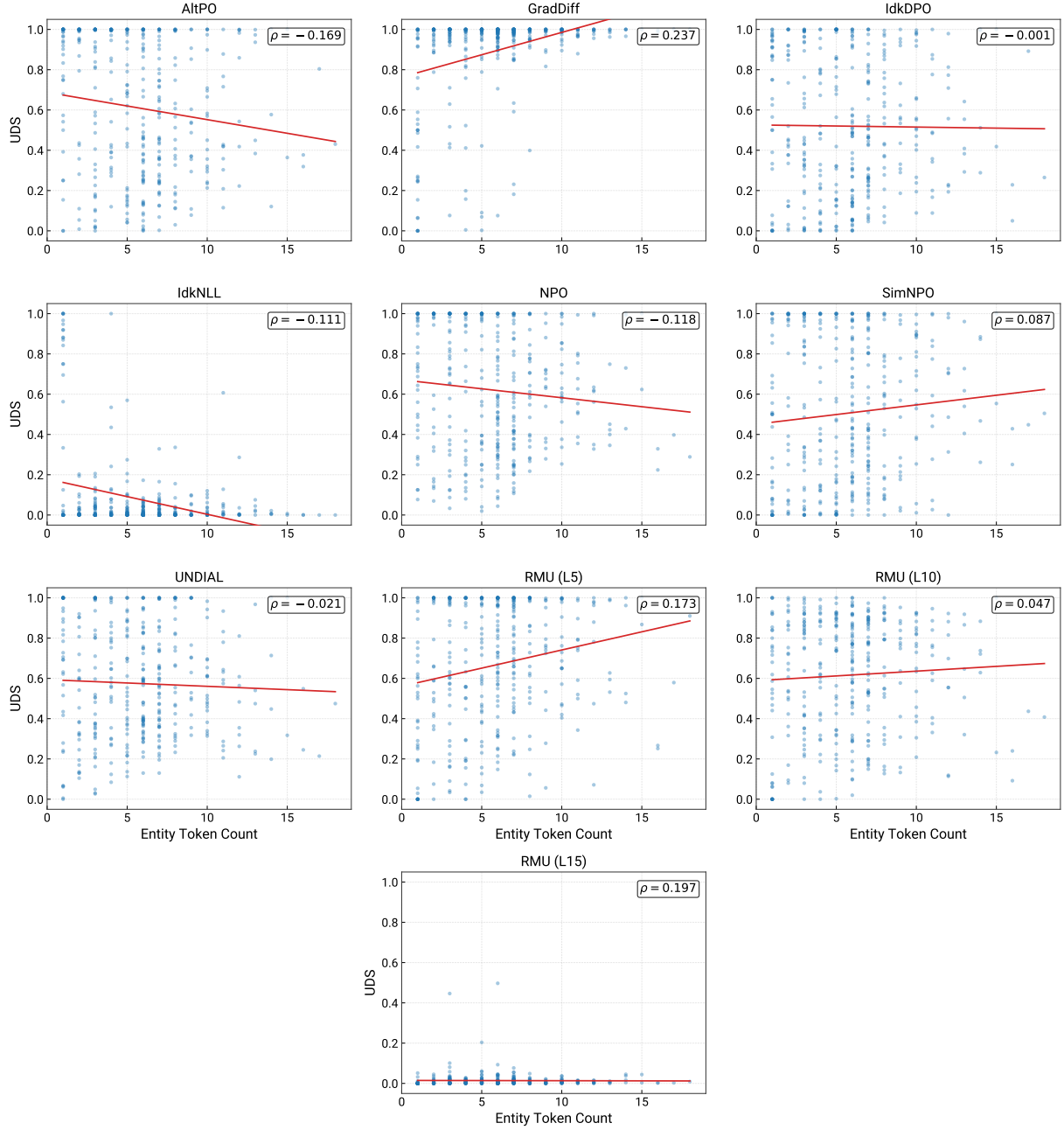


Figure 6: Per-example UDS vs. entity token count ($\text{lr}=2\text{e-}5$, epoch 10; UNDIAL uses $\text{lr}=1\text{e-}4$). RMU variants differ by the target layer l at which the steering loss is applied (L5, L10, L15). All methods show $|\rho| < 0.24$ with mixed sign, indicating no consistent directional bias.

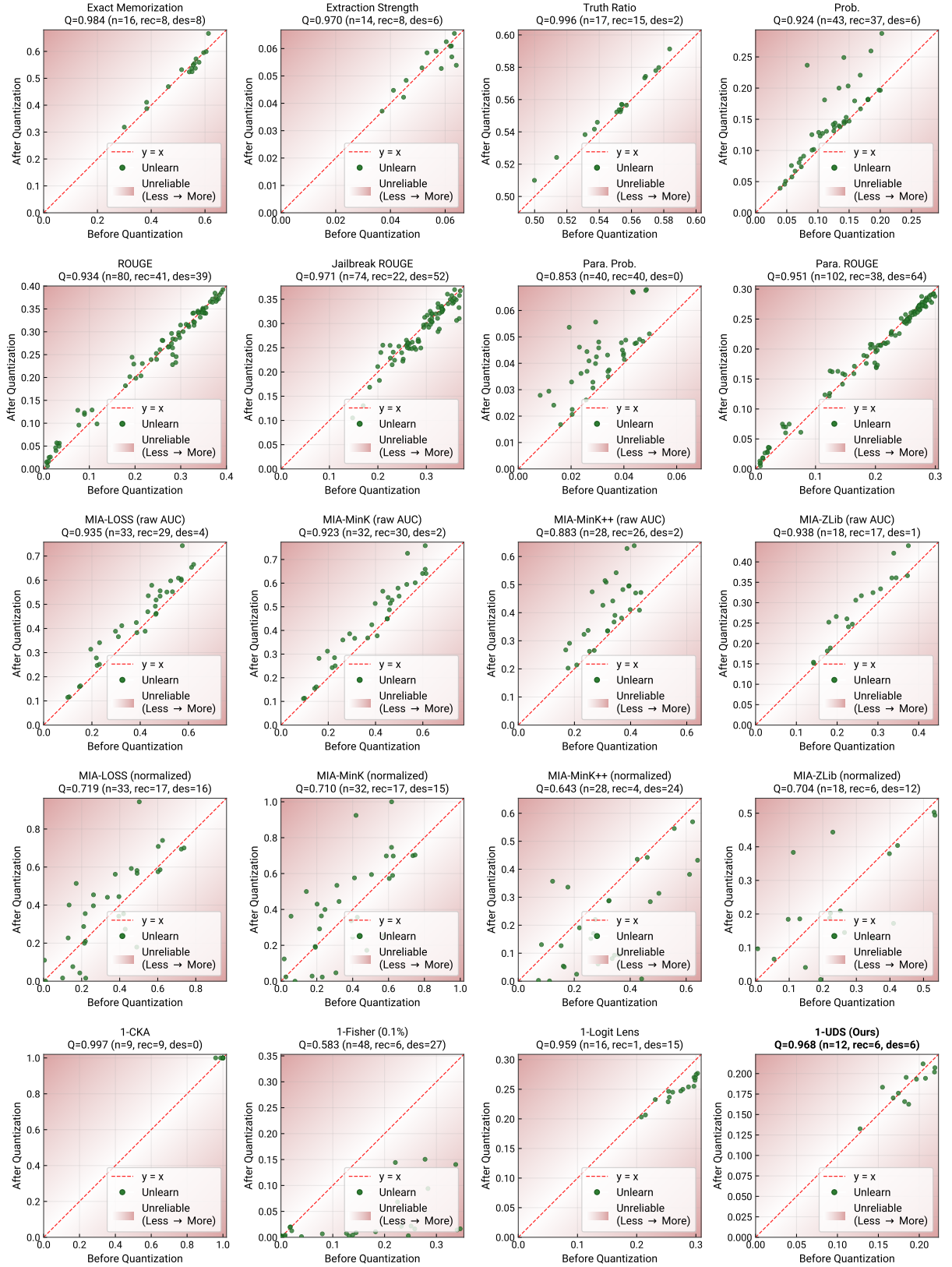


Figure 7: Quantization robustness (Q) for all 20 metrics after utility and faithfulness filtering. Each subplot plots the metric value before (x) vs. after (y) NF4 4-bit quantization, with the number of models showing recovery (rec) or destruction (des). n is the number of models that passed both filters for that metric. The background gradient indicates deviation from the $y = x$ reference: white = stable, red = unstable.

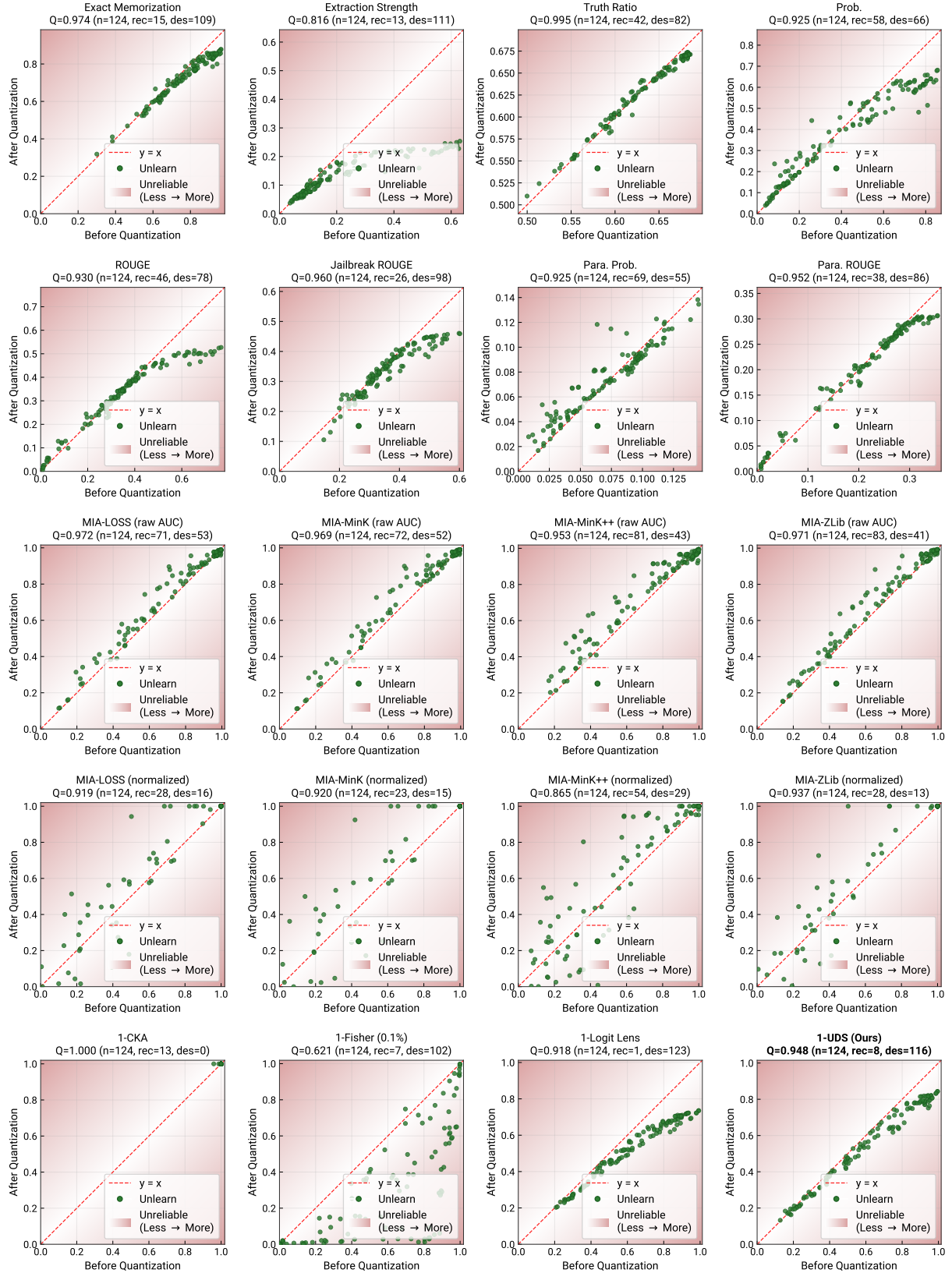


Figure 8: Quantization robustness (Q) for all 20 metrics after utility filtering only. Each subplot plots the metric value before (x) vs. after (y) NF4 4-bit quantization, with the number of models showing recovery (rec) or destruction (des). n is the number of models that passed the utility filter for that metric. The background gradient indicates deviation from the $y = x$ reference: white = stable, red = unstable. Metrics such as Extraction Strength and ROUGE show score drops across many models after quantization.

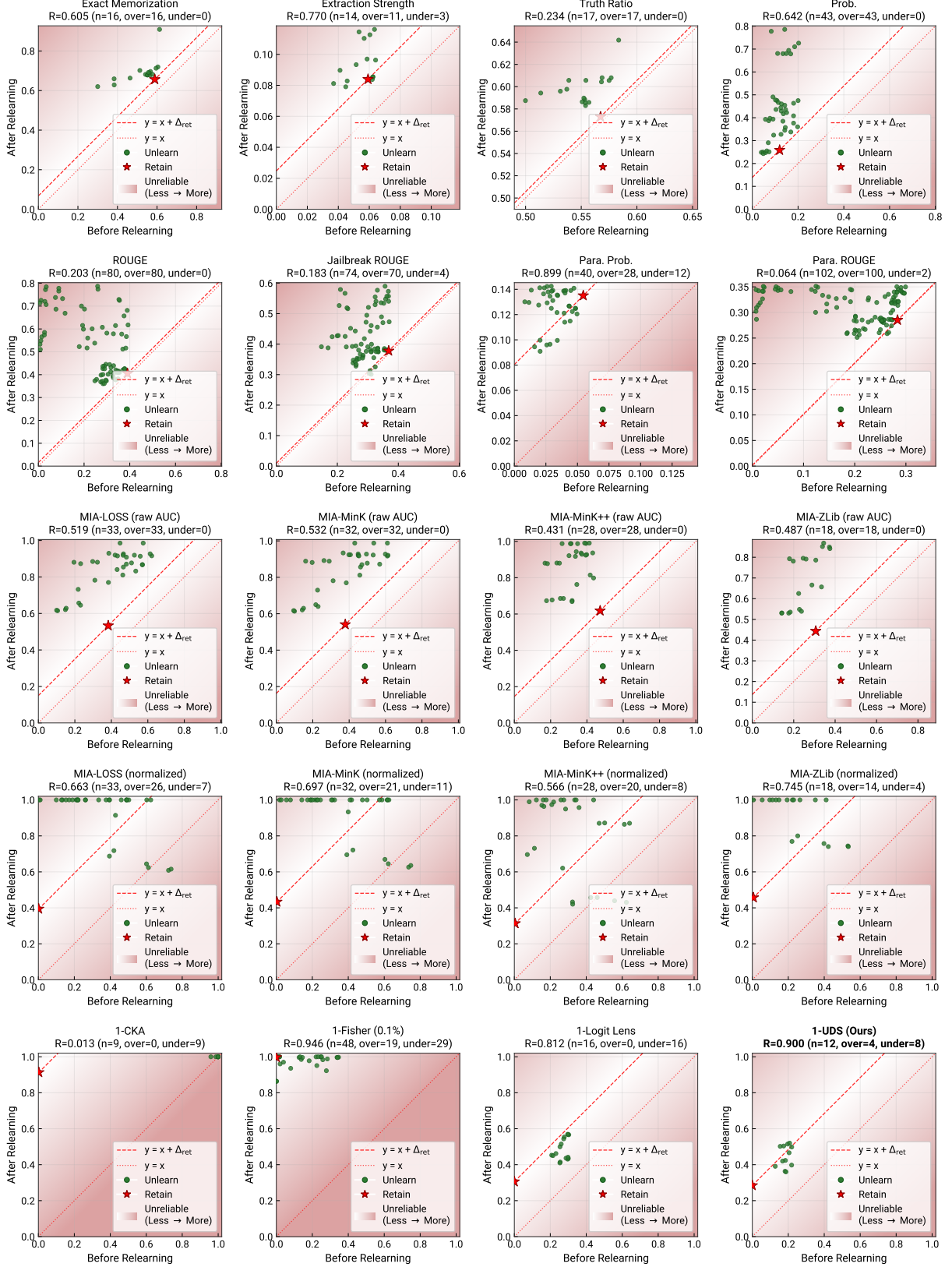


Figure 9: Relearning robustness (R) for all 20 metrics after utility and faithfulness filtering. Each subplot plots the metric value before (x) vs. after (y) one epoch of relearning, with the number of models showing over-recovery (over) or under-recovery (under). n is the number of models that passed both filters for that metric. The dashed line shows $y = x + \Delta_{\text{ret}}$ (expected behavior given the retain model’s shift); the dotted line shows $y = x$. The background gradient indicates deviation from the expected line: white = stable, red = unstable. The red star marks the retain model.