Forgetting Only What Matters: Layer-Selective Unlearning toward Robust LLMs

Ravi Ranjan *
Florida International University (FIU), Miami, FL, USA
rkuma031@fiu.edu

Olivera Kotevska
Oak Ridge National Laboratory (ORNL), Oak Ridge, TN, USA
kotevskao@ornl.gov

Agoritsa Polyzou
Florida International University (FIU), Miami, FL, USA
apolyzou@fiu.edu

Abstract

Large Language Models (LLMs) can memorize and reproduce sensitive, copyrighted, or otherwise undesirable training content, creating privacy, safety, and regulatory concerns. Machine unlearning offers a practical alternative to full retraining, but many existing methods apply broad or fixed parameter updates that can degrade utility and remain brittle under deployment changes such as post-training quantization, where forgotten knowledge may partially re-emerge. We propose ***F**orgetting **O**nly What **M**atters via **U**nlearning **L**ayers (**FOM-UL**)*, a layer-level unlearning framework that selects transformer layers using a forget-to-retain significance score. This score identifies layers with high influence on the forget set and low sensitivity to the retain set, allowing FOM-UL to concentrate updates where they are most effective while leaving most of the model unchanged. This targeted update strategy improves the forgetting-utility trade-off and provides an empirical path toward quantization-resilient unlearning by reducing the chance that small, diffuse updates are erased by low-bit rounding. Across TOFU, KnowUnDo, and MUSE-style evaluations, FOM-UL reduces residual memorization compared with strong baselines while preserving retain-set utility close to the vanilla model. Under 8-bit and 4-bit post-training quantization, FOM-UL maintains a favorable forgetting-utility trade-off, combining competitive memorization suppression with stronger retained utility. Overall, FOM-UL provides an efficient unlearning strategy that improves targeted forgetting, utility preservation, and deployment robustness without claiming formal guarantees of erasure.
Code available at:
https://github.com/raviranjan-ai/FOMUL-AACL-2026.

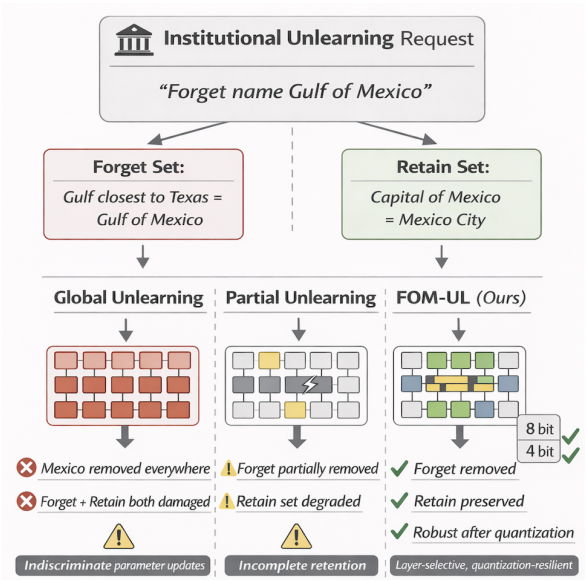


Figure 1: FOM-UL robust and quantization-resilient forgetting against global and partial unlearning methods.

1 Introduction

Large Language Models (LLMs) have transformed natural language processing, delivering strong performance across diverse tasks and domains (Zhao et al., 2023). Yet, as these models scale and are deployed widely, an increasingly visible failure mode is their tendency to memorize and reproduce fragments of training data, including sensitive personal information, copyrighted text, or otherwise undesirable content (Zhao et al., 2023; Wang et al., 2025a). Such memorization creates legal, ethical, and security risks, especially in high-stakes settings where accidental disclosure of protected content is unacceptable. These concerns are further amplified by regulatory requirements such as the General Data Protection Regulation (GDPR) "right to be forgotten" (Council et al., 2022), which demands mechanisms to remove specific data upon request.

Machine unlearning for LLMs has therefore emerged as a practical alternative to full retraining: the goal is to remove targeted knowledge or

*Corresponding author.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 4738–4760
November 6-10, 2026 ©2026 Association for Computational Linguistics

behaviors from a trained model while preserving its overall capabilities. However, existing unlearning pipelines face major obstacles. Retraining is often prohibitively expensive at LLM scale (Jang et al., 2022), and frequent unlearning requests in real-world deployments further stress the need for efficient, deployable solutions. Beyond cost, the high dimensionality and tightly coupled representations of modern transformer architectures make targeted knowledge removal inherently difficult: even seemingly localized edits can be propagated broadly, causing utility degradation or catastrophic forgetting (Zhang et al., 2024a). These challenges have motivated a rapidly growing literature on LLM unlearning across widely used architectures such as GPT-2, LLaMA, and Gemma (Geng et al., 2025; Yao et al., 2024a).

A central issue is that many unlearning methods still rely on global or otherwise indiscriminate parameter updates, which can be brittle and imprecise. Figure 1 contrasts this paradigm with our proposed approach: rather than updating large portions of the model, **Forgetting Only What Matters via Unlearning Layers (FOM-UL)** selectively modifies only those transformer layers most responsible (high forget-to-retain ratio) for encoding the undesired knowledge. This targeted intervention aims to improve approximate forgetting while preserving retain-set utility and minimizing collateral damage. *Targeted unlearning* differs from general-purpose model editing by focusing on the selective removal of *specific* facts, documents, or behaviors, enabling an LLM to "forget" designated content without retraining from scratch. Importantly, unlearning is not standard fine-tuning in reverse: whereas fine-tuning typically reinforces desired behavior via positive examples, unlearning must suppress undesired behavior.

A range of approaches have been proposed to realize this objective (Geng et al., 2025). In particular, preference-optimization (PO) and gradient-ascent (GA) style methods are widely adopted due to their simplicity and effectiveness (Liu et al., 2024a). PO-based methods cast unlearning as an alignment problem in which the model is steered to prefer alternative responses using preference pairs. Complementary directions, including relabeling, adapter/LoRA-based updates, quantization-based techniques (Zhang et al., 2024c), and reinforcement learning (Lu et al., 2022)-provide additional tools, but PO and GA remain central to scalable unlearning.

Gradient Ascent (GA) is a common unlearning strategy that increases the loss on memorized responses to reduce confidence in targeted outputs (Yao et al., 2024b). However, GA typically applies broad model-wide updates, which can leave residual memorization and harm unrelated capabilities (Yao et al., 2024a), and it is often brittle under post-training quantization, where discretizations may partially restore suppressed behaviors. Although retain-set regularization (Liu et al., 2024a) and KL-divergence constraints to the original model (Yao et al., 2024a) help preserve utility, they do not fully eliminate the side effects of indiscriminate updates. Motivated by recent evidence of quantization-induced failure modes in unlearning (Zhang et al., 2024b), we instead pursue a layer-selective strategy that concentrates updates where they matter most, improving precision and robustness against quantization-driven recovery.

FOM-UL is a targeted and efficient unlearning framework that mitigates catastrophic forgetting, quantization-induced relearning, and the inefficiency of global parameter updates. Our primary contributions are:

(i) Layer-level forget-retain localization. We introduce a layer-selection criterion based on the ratio between forget-set gradient magnitude and retain-set gradient magnitude. This identifies layers that offer high forgetting leverage with comparatively low retain-set interference.

(ii) Iterative layer-budget expansion. Rather than updating a fixed region or the full model, FOM-UL starts from a small high-score layer set and expands it only when forgetting criteria are unmet, improving the trade-off between erasure strength and utility preservation.

(iii) Efficient selective optimization. By restricting updates to a small number of selected layers, FOM-UL reduces trainable parameters, memory use, and runtime while remaining compatible with standard GA, NPO, and KLD-style unlearning losses.

(iv) Empirical quantization-robustness analysis. Motivated by quantization-induced recovery, we test FOM-UL under 8-bit and 4-bit post-training quantization and show that concentrated layer updates reduce residual memorization compared with global or fixed-selection baselines.

2 Preliminary and Related Work

Machine unlearning in Large Language Models (LLMs) involves selectively removing specific learned knowledge without significantly degrading overall model performance (Geng et al., 2025; Liu et al., 2025a; Jang et al., 2022; Huang et al., 2024). Formally, given a model f_θ parameterized by θ and a dataset D_{forget} representing undesirable knowledge, the parameter updates of machine unlearning can be expressed as:

$$\theta' = \theta + \eta \nabla_\theta \mathcal{L}_{\text{forget}}(\theta, D_{\text{forget}}), \tag{1}$$

where η is the learning rate and $\mathcal{L}_{\text{forget}}$ is typically a loss function defined on the forget set, often optimized via gradient ascent (GA) (Bourtoule et al., 2021; Golatkar et al., 2020).

Recent advances in LLM unlearning have established several effective approaches to remove specific knowledge or capabilities from trained models. Parameter-based methods modify model weights directly through techniques, like gradient ascent (Jang et al., 2022) or localized weight editing (Ilharco et al., 2022). Global unlearning updates the full parameter space of an LLM, which can aggressively suppress targeted behavior but often propagates changes widely, leading to broader utility degradation and higher computational cost (Wuerkaixi et al., 2025). In contrast, partial unlearning restricts updates to a subset of components (e.g., layers, modules, or adapters) to limit collateral damage and improve efficiency (Wang et al., 2025b), yet it can suffer from *incomplete forgetting* when the targeted knowledge is distributed across multiple parts of the network (Yao et al., 2024a).

Knowledge-boundary methods create negative examples to teach models to avoid certain responses (Li et al., 2025; Liu et al., 2024b). Contrastive unlearning pairs forgetting targets with similar but acceptable content to refine decision boundaries (Chen and Yang, 2023). Dataset-filtering approaches reconstruct training data while excluding unwanted information (Zhao et al., 2025). Optimization-based methods, like Influence Tuning, use importance scores to identify and modify critical parameters (Xu et al., 2024). Direct Preference Optimization (DPO) is a prominent example that assigns higher preference to neutral or refusal outputs than to the original memorized responses (Rafailov et al., 2023). Negative Preference Optimization (NPO) further streamlines this process by relying only on negative forget samples (Zhang et al., 2024a). These approaches vary in their effectiveness, computational requirements, and ability to preserve model performance on unrelated tasks.

However, many unlearning methods remain vulnerable to *quantization-induced relearning*, where low-bit quantization can effectively erase small unlearning updates and restore behavior close to the original model. Post-training quantization is widely used to reduce the computational and storage overhead of LLMs by mapping full-precision parameters to low-bit representations (Gholami et al., 2022; Lin et al., 2024). Recent quantization methods include 8-bit and 4-bit techniques, significantly enhancing inference efficiency while preserving acceptable accuracy levels (Dettmers et al., 2022). Nevertheless, quantization can adversely affect machine unlearning performance, potentially exacerbating the issue of catastrophic forgetting (Luo et al., 2023; Ranjan et al., 2026).

Formally, let θ be the original parameters and θ' the post-unlearning parameters, and let $Q(\cdot)$ denote post-training rounding quantization (element-wise or per-group). For step size Δ_j on coordinate (group) j, we have

$$Q_{\Delta_j}(w) = \Delta_j \,\text{Round}\left(\frac{w}{\Delta_j}\right). \tag{2}$$

Quantization can mask an unlearning update when the pre- and post-unlearning parameters remain in the same quantization bin. For coordinate (or group) j, this occurs exactly when

$$\begin{aligned} Q_{\Delta_j}(\theta'_j) = Q_{\Delta_j}(\theta_j) \iff \\ \text{Round}\left(\frac{\theta'_j}{\Delta_j}\right) = \text{Round}\left(\frac{\theta_j}{\Delta_j}\right). \end{aligned} \tag{3}$$

If this condition holds for a large fraction of the edited coordinates, the quantized unlearned model can become functionally closer to the quantized original model, allowing part of the suppressed knowledge to re-emerge (Zhang et al., 2024b). Unlike prior fixed forget-saliency masks, FOM-UL jointly uses forget-to-retain gradient ratios, adaptive layer-budget expansion, and multi-objective selective optimization.

Another prominent line of work leverages parameter-efficient modules, e.g., adapters and LoRA layers, to localize updates and preserve the bulk of pre-trained weights (Liu et al., 2025b; Li and Liang, 2021; Hu et al., 2022). Moreover, even after "erasure", residual information

can be resurrected by adversarial or carefully engineered prompts, so-called spillage, undermining any privacy guarantees (Ji et al., 2024; Zhang et al., 2024b).

In summary, while existing methods, such as quantization-based unlearning, knowledge editing, and parameter-efficient tuning, have provided valuable insights into model modification, they exhibit significant limitations when applied to robust unlearning scenarios.

3 Proposed Method: FOM-UL

We propose FOM-UL, a targeted framework that suppresses specific knowledge by updating a small set of layers with high forget-to-retain significance. Rather than claiming exact erasure, FOM-UL aims to approximate retraining behavior on the forget set while preserving retain-set utility and improving robustness to quantization-induced recovery.

Problem Formulation. Let $\mathcal{M}_\theta$ denote a pretrained transformer-based model with parameters $\theta \in \mathbb{R}^d$. Given two datasets a forget set D_{forget}, containing the data to be erased, and a retain set D_{retain}, comprising knowledge that must be preserved the objective is to update the model parameters to minimize verbatim memorization (VerMem) and privacy leakage (PrivLeak) on D_{forget}, while preserving utility and knowledge retention on D_{retain}.

Motivation. Modern LLMs, such as Llama-2, are typically *decoder-only* autoregressive transformers with L stacked layers. For an input prefix $x_{1:t}$, each layer ℓ updates hidden states $h_{1:t}^{(\ell)}$ via (i) *multi-head self-attention* (MHSA) and (ii) a position-wise *feed-forward network* (FFN), coupled with residual connections and normalization. In MHSA, each *attention head* performs a query-key-value interaction to form a weighted mixture of contextual token representations, and the layer aggregates diverse dependency patterns (e.g., local syntax and long-range factual cues) across heads. Importantly, information is *not uniformly distributed* across the network: lower layers tend to encode lexical/syntactic features, intermediate layers increasingly represent semantic relations, and deeper layers are more directly coupled to the final *logits* (next-token scores) used for generation. Consequently, memorized or sensitive content can be disproportionately concentrated in a subset of layers/heads that exert outsized influence on specific next-token predictions. This heterogeneous localization motivates our work. By attributing a forget objective to the layers that most affect the target prediction and selectively updating only those layers, FOM-UL can induce targeted forgetting with reduced collateral utility loss, while producing parameter shifts that are more resilient to quantization-induced reversal than diffuse, global updates.

Proposed Method Overview. Figure 2 presents the complete FOM-UL pipeline. First, we select the datasets, a *Forget* set (e.g., copyrighted Harry Potter text) and a *Retain* set (e.g., fandom wiki and general knowledge). Next, a layer-attribution analysis identifies the transformer layers most responsible for encoding the sensitive content. A binary saliency mask is then generated (for n layers) to *freeze* layers with low attribution and *unlock* only the high-attribution layers for updates. Targeted unlearning alternates between (i) gradient ascent on the Forget set to maximize loss on unwanted content and (ii) gradient descent on the Retain set to preserve essential knowledge, applied solely within the selected layers. Finally, the resulting model is evaluated on verbatim memorization, privacy leakage, knowledge retention, and general utility to verify robust, quantization-resilient forgetting.

Identifying Key Layers. We consider a transformer-based LLM with parameters $\theta = \{\theta^{(1)}, \ldots, \theta^{(L)}\}$ grouped by layer, and let $x_{1:t}$ denote an input prefix. We define $w_{t+1}^* = \arg\max_w P_\theta(w \mid x_{1:t})$ as the model's top-1 next-token prediction and compute its baseline probability

$$\hat{p} \;=\; P_\theta\big(w_{t+1}^* \mid x_{1:t}\big)\,. \tag{4}$$

To localize where this prediction is formed, we perform the following layer-wise ablation: for each layer $\ell \in \{1, \ldots, L\}$, we intervene on the forward pass by removing (e.g., zeroing) the contribution of layer ℓ to the residual stream, and recompute the same next-token probability under this ablated model, denoted by $P_{\theta \setminus \ell}$:

$$\hat{p}_\ell \;=\; P_{\theta \setminus \ell}\big(w_{t+1}^* \mid x_{1:t}\big)\,. \tag{5}$$

This demonstrates that prediction influence is heterogeneous across transformer layers. However, FOM-UL does not use $\hat{p}_\ell$ for optimization or layer selection. Instead, the trainable layer set and its iterative expansion are determined by the forget-to-retain significance score $\mathrm{Sig}(\ell)$ in Eq. 8.

To prioritize layers that enable effective forgetting *with minimal retain-set disruption*, FOM-UL further computes a gradient-based *significance*

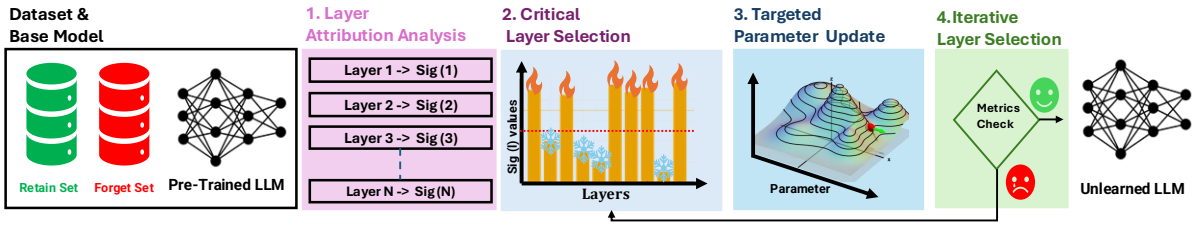


Figure 2: Overview of the Forgetting Only What Matters via Unlearning Layers (FOM-UL) framework.

score. Let $\mathcal{L}_{\text{forget}}(\theta; B_f)$ and $\mathcal{L}_{\text{retain}}(\theta; B_r)$ be the losses on forget and retain batches $B_f \subset \mathcal{D}_f$ and $B_r \subset \mathcal{D}_r$, respectively. We define the per-layer gradient magnitudes

$$I(\ell) \;=\; \left\|\nabla_{\theta^{(\ell)}} \mathcal{L}_{\text{forget}}(\theta; B_f)\right\|_2, \tag{6}$$

$$I_r(\ell) \;=\; \left\|\nabla_{\theta^{(\ell)}} \mathcal{L}_{\text{retain}}(\theta; B_r)\right\|_2, \tag{7}$$

and the normalized forget-to-retain trade-off score as the **significance score** used to rank the layers:

$$\text{Sig}(\ell) \;=\; \frac{I(\ell)}{I_r(\ell) + \varepsilon}, \tag{8}$$

where $\varepsilon > 0$ is a small constant for numerical stability. Intuitively, high $\text{Sig}(\ell)$ identifies layers that are highly responsive to the forgetting objective while being comparatively insensitive to the retain objective. In the first stage, FOM-UL selects the candidate set

$$S \;=\; \{\ell \in \{1, \ldots, L\} \;:\; \text{Sig}(\ell) > \tau\}, \tag{9}$$

where τ is a tunable threshold controlling the layer budget and the conservativeness of the update set (detailed in Appendix D).

Selective Layer Update with Forget, Mismatch, and Retain Losses. FOM-UL performs targeted updates only on layers $\ell \in S$. The update is guided by three loss components (Detailed in Appendix D.2): (i) the forgetting loss $\mathcal{L}_{\text{forget}}$ to remove memorized patterns, (ii) the mismatch loss $\mathcal{L}_{\text{mismatch}}$ to diverge from original outputs, and (iii) the retain loss $\mathcal{L}_{\text{retain}}$ to preserve general utility. For each selected layer, the parameter update is:

$$\begin{aligned} \theta_{t+1}^{(\ell)} = \theta_t^{(\ell)} + \eta_F \nabla_{\theta^{(\ell)}} \mathcal{L}_{\text{forget}} \\ + \eta_M \nabla_{\theta^{(\ell)}} \mathcal{L}_{\text{mismatch}} - \eta_R \nabla_{\theta^{(\ell)}} \mathcal{L}_{\text{retain}}, \end{aligned} \tag{10}$$

where η_F, η_M, and η_R are the respective learning rates. Layers not in S remain frozen:

$$\theta_{t+1}^{(\ell)} = \theta_t^{(\ell)}, \;\; \forall \ell \notin S. \tag{11}$$

Iterative Expansion and Stopping Criteria. FOM-UL proceeds iteratively: after an initial selection of top k layers, ranked by $\text{Sig}(\ell)$, that cross the threshold τ, if the forgetting objectives (e.g., VerMem below threshold < 0.05) are unmet, the set S is expanded by adding the next most significant layer based on an additional significance scoring, i.e., $S' = S \cup \{arg\max_{\ell \notin S} \text{Sig}(\ell)\}$. This continues until the forgetting metric converges or a maximum number of epochs is reached. This prevents aggressive updates early on and reduces the risk of unintended utility loss. Extended justifications and details can be found in Appendix D.3.

The pseudocode is provided in Appendix A, additional methodological details are described in Appendix D, and formal justifications for the proposed layer selection strategy (Lemma 1) and iterative unlearning procedure (Lemma 2) are presented in Appendix E.

4 Experiments

4.1 Experimental Setup

Detailed implementation details, hyperparameters, and metric definitions are provided in Appendix B and Appendix C.

Baselines. We compare FOM-UL against the vanilla model and a broad set of LLM unlearning baselines. Following the SURE protocol (Zhang et al., 2024b), we evaluate Gradient Ascent (GA) and Negative Preference Optimization (NPO), combined with common retain-preserving objectives such as gradient descent on the retain set (GDR) and KL regularization (KLR). GA directly reduces confidence on forget samples, while NPO treats forget examples as negative preferences. We also include recent state-of-the-art methods, including ReLearn (Xu et al., 2025), which performs unlearning through data augmentation and fine-tuning, and LUNAR (Shen et al., 2025), which redirects internal activations. In addition, we consider parameter-efficient and quantization-aware unlearning baselines to evaluate robustness and efficiency.

Table 1: **Main unlearning results on TOFU-World Facts.** Results are reported as mean±std over three run. M1/M2 are ROUGE-based residual memorization scores, M3 measures privacy leakage distance to the retrained model, and M4 measures retain-set utility. Values are reported on a compact 0–10 scale for readability; multiplying by 10 converts them to a 0–100 scale.

Method	GPT-2				Llama-3.2-1B				Gemma-3-1B			
	M1↓	M2↓	M3→ 0	M4↑	M1↓	M2↓	M3→ 0	M4↑	M1↓	M2↓	M3→ 0	M4↑
Vanilla Model	9.00±0.08	6.10±0.06	9.50±0.12	**2.90±0.03**	5.80±0.07	5.80±0.07	8.80±0.15	**2.90±0.03**	5.60±0.06	5.60±0.06	8.50±0.14	**2.90±0.03**
GA_{GDR}	4.47±0.10	4.49±0.10	5.04±0.15	2.07±0.04	4.24±0.09	4.24±0.09	4.88±0.13	2.07±0.04	4.16±0.09	4.16±0.09	4.80±0.13	2.14±0.04
SURE + GA	2.14±0.06	2.16±0.06	4.44±0.12	1.96±0.04	2.08±0.05	2.08±0.05	4.24±0.11	1.98±0.04	2.06±0.05	2.06±0.05	4.20±0.11	1.98±0.04
FOM-UL + GA	3.02±0.07	3.06±0.07	3.08±0.10	2.34±0.03	2.96±0.06	2.98±0.06	2.92±0.09	2.42±0.03	2.94±0.06	2.96±0.06	2.88±0.09	2.42±0.03
NPO_{GDR}	2.14±0.08	2.16±0.08	8.10±0.18	2.07±0.04	2.06±0.07	2.06±0.07	6.94±0.18	2.36±0.04	2.06±0.07	2.06±0.07	6.60±0.17	2.40±0.04
SURE + NPO	1.60±0.05	1.62±0.05	3.64±0.11	1.92±0.04	1.60±0.04	1.60±0.04	3.40±0.10	1.96±0.04	1.60±0.04	1.60±0.04	3.54±0.10	1.96±0.04
FOM-UL + NPO	1.68±0.05	1.70±0.05	2.84±0.09	2.28±0.04	1.58±0.04	1.60±0.04	2.66±0.08	2.34±0.04	1.58±0.04	1.60±0.04	2.60±0.08	2.38±0.04
KLD	4.47±0.11	4.52±0.11	7.92±0.18	1.92±0.05	4.04±0.10	4.04±0.10	6.00±0.16	1.98±0.04	4.00±0.10	4.00±0.10	5.60±0.16	1.98±0.04
SURE + KLD	1.96±0.05	1.98±0.05	4.00±0.12	2.16±0.04	1.90±0.04	1.90±0.04	3.84±0.11	2.12±0.04	1.88±0.04	1.88±0.04	3.80±0.11	2.12±0.04
FOM-UL + KLD	2.74±0.07	2.78±0.07	3.18±0.10	2.30±0.04	2.62±0.06	2.64±0.06	3.04±0.09	2.36±0.04	2.60±0.06	2.62±0.06	3.00±0.09	2.36±0.04
ReLearn	5.24±0.12	5.36±0.12	5.00±0.14	2.00±0.04	5.04±0.11	5.04±0.11	4.80±0.13	2.14±0.04	5.02±0.11	5.02±0.11	4.80±0.13	2.14±0.04
LUNAR	**1.54±0.04**	1.60±0.04	4.24±0.12	1.86±0.05	1.28±0.03	1.28±0.03	4.10±0.11	2.06±0.04	1.22±0.03	**1.22±0.03**	4.10±0.11	2.06±0.04
FOM-UL-Full (Ours)	1.56±0.03	**1.58±0.03**	**2.42±0.07**	2.84±0.03	**1.24±0.03**	**1.24±0.03**	**1.96±0.06**	**2.90±0.03**	**1.20±0.03**	1.24±0.03	**1.92±0.06**	**2.90±0.03**

Datasets. We evaluate on three standard LLM unlearning benchmarks. TOFU (Maini et al., 2024) tests factual QA forgetting over synthetic world facts. KnowUnDo (Tian et al., 2024) evaluates privacy- and copyright-oriented unlearning while checking whether useful knowledge is unintentionally removed. We also use MUSE (Shi et al., 2025), including BOOKS and NEWS. BOOKS uses the Harry Potter corpus as the forget set and FanWiki as the retain set, while NEWS contains BBC articles split into forget, retain, and holdout subsets for evaluating memorization, utility, and privacy leakage.

Metrics. We follow the standard four-metric protocol used in prior work (Zhang et al., 2024b). M1 Verbatim Memorization and M2 Knowledge Memorization measure residual content from the forget set; lower values indicate stronger forgetting. M3 Privacy Leakage measures membership-inference risk and is best when close to zero. M4 Utility Preservation measures retained knowledge on the retain set, where higher values indicate better utility. Together, these metrics capture the main trade-off between erasing unwanted knowledge and preserving useful behavior.

Models. We evaluate FOM-UL on multiple transformer-based LLMs, including Llama-2 7B, Llama-3.2 1B (Grattafiori et al., 2024), GPT-2 (Hanna et al., 2023), and Gemma-3 1B (Team et al., 2024). These models cover different scales and architectures, allowing us to test whether layer-selective unlearning remains effective across model families.

4.2 Unlearning Results

Performance Comparison. Table 1 shows two key findings. First, FOM-UL consistently achieves the best overall forgetting-utility balance across GPT-2, Llama-3.2-1B, and Gemma-3-1B, reducing residual memorization and privacy leakage while keeping retain-set utility close to the vanilla model. Second, the gains are stable across different base objectives, showing that the proposed layer-selection strategy improves standard GA, NPO, and KLD-style unlearning rather than depending on a single loss. Based on these results, Table 2 evaluates the best-performing combinations across NEWS, KnowUnDo, and BOOKS.

Evaluation of the Best-Performing combinations. Table 2 shows that FOM-UL provides the strongest overall forgetting-utility trade-off across NEWS, KnowUnDo, and BOOKS: it matches or ties the best memorization scores while substantially reducing privacy leakage. It also preserves retain-set utility close to the vanilla model, indicating that the gains are not due to destructive over-unlearning.

4.3 Robustness Analysis.

Adversarial/Jailbreak Robustness. As shown in Figure 3, FOM-UL achieves the lowest ALR of 11.6%, compared to 16.5% for SURE+NPO and 19.8% for LUNAR. This indicates stronger resistance to jailbreak-based recovery of forgotten knowledge. These results show that FOM-UL remains effective not only under clean prompts, but also under adversarial extraction attempts. The complete results are provided in Appendix F.

Table 2: Best-performing unlearning combinations on Llama-3.2-1B across three datasets. Results are reported as mean±std over three runs.

Method	NEWS				KnowUnDo				BOOKS			
	M1↓	M2↓	M3→ 0	M4↑	M1↓	M2↓	M3→ 0	M4↑	M1↓	M2↓	M3→ 0	M4↑
Vanilla Model	5.80±0.07	5.80±0.07	8.80±0.15	2.90±0.03	5.80±0.07	5.80±0.07	8.80±0.15	2.90±0.03	5.80±0.07	5.80±0.07	8.80±0.15	2.90±0.03
GA_{GDR}	4.24±0.09	4.24±0.09	4.84±0.13	2.10±0.04	4.26±0.09	4.26±0.09	4.82±0.13	2.10±0.04	4.24±0.09	4.24±0.09	4.84±0.13	2.10±0.04
NPO_{GDR}	2.16±0.07	2.16±0.07	8.00±0.18	2.10±0.04	2.12±0.07	2.12±0.07	7.40±0.17	2.10±0.04	2.16±0.07	2.16±0.07	8.00±0.18	2.10±0.04
KLD	4.52±0.10	4.52±0.10	7.24±0.17	1.98±0.04	4.26±0.10	4.26±0.10	6.86±0.16	2.00±0.04	4.52±0.10	4.52±0.10	7.24±0.17	1.98±0.04
SURE + NPO	1.62±0.02	1.62±0.04	3.68±0.10	1.98±0.04	1.64±0.03	1.64±0.04	3.72±0.10	2.00±0.04	1.62±0.05	1.62±0.04	3.68±0.10	1.98±0.04
ReLearn	5.22±0.11	5.22±0.11	4.96±0.13	2.08±0.04	5.26±0.11	5.26±0.11	4.96±0.13	2.10±0.04	5.22±0.11	5.22±0.11	4.96±0.13	2.08±0.04
LUNAR	**1.56±0.03**	**1.56±0.03**	4.20±0.11	2.14±0.04	1.60±0.03	1.60±0.03	4.20±0.11	2.16±0.04	**1.56±0.03**	**1.56±0.03**	4.20±0.11	2.14±0.04
FOM-UL	**1.56±0.03**	**1.56±0.02**	**2.64±0.07**	**2.90±0.02**	**1.58±0.03**	**1.58±0.03**	**2.64±0.08**	**2.90±0.03**	**1.56±0.04**	**1.56±0.03**	**2.64±0.08**	**2.90±0.03**

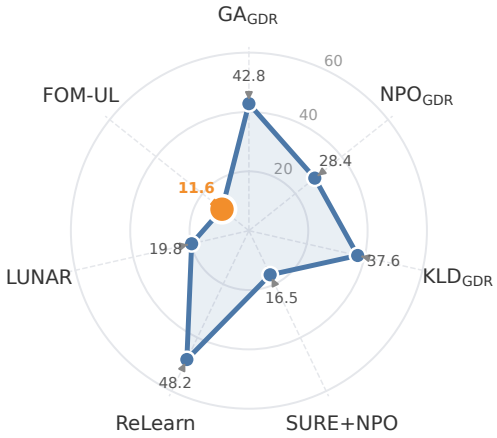


Figure 3: Attack Leakage Rate (ALR) under adversarial/jailbreak prompts. Lower ALR indicates fewer successful recoveries of forgotten on Llama-3 and TOFU dataset under adversarial prompting; FOM-UL achieves the lowest leakage score among all methods (Table 9).

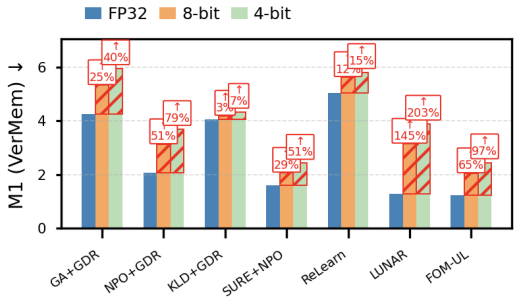


Figure 4: Quantization robustness of unlearning methods measured by M1 (VerMem, lower is better) under Full precision 32-bit (FP-32), 8-bit, and 4-bit post-training quantization; red annotations denote the relative M1 increase when moving from FP-32 to 8-bit and 4-bit.

Quantization Robustness. Table 3 shows that 4-bit quantization generally weakens unlearning by increasing residual memorization relative to 8-bit precision. FOM-UL reduces residual memorization and improves privacy-parity metrics relative to retraining. However, its negative M3 values indicate deviation from the retrained privacy baseline rather than improved privacy. We therefore interpret M3 by distance to zero and report these values as evidence of privacy-behavior shift under aggressive quantization, while the main robustness gain of FOM-UL is strongest on memorization and utility. Additional analysis is provided in Appendices C and D.4.

We also evaluate the performance of the proposed method on Qwen3.5-4B (Team, 2026) using

Table 3: Quantization robustness on Llama-3.2-1B using TOFU-World Facts. Results compare 8-bit and 4-bit post-training quantization.

Method.	Quant.	M1↓	M2↓	M3→ 0	M4↑
8-bit Quantization					
GA_{GDR}	8-bit	5.32	5.32	4.92	1.94
NPO_{GDR}	8-bit	3.12	3.12	6.98	2.04
KLD_{GDR}	8-bit	4.18	4.18	6.18	1.82
SURE + NPO	8-bit	2.06	2.06	3.62	1.84
ReLearn	8-bit	5.62	5.62	4.84	1.92
LUNAR	8-bit	3.14	3.14	4.96	1.72
FOM-UL	8-bit	**2.04**	**2.04**	**−1.47**	**2.43**
4-bit Quantization					
GA_{GDR}	4-bit	5.94	5.94	5.08	1.67
NPO_{GDR}	4-bit	3.68	3.68	7.02	2.00
KLD_{GDR}	4-bit	4.32	4.32	6.22	1.80
SURE + NPO	4-bit	**2.42**	**2.42**	3.68	1.80
ReLearn	4-bit	5.80	5.80	4.92	1.87
LUNAR	4-bit	3.88	3.88	5.16	1.70
FOM-UL	4-bit	2.44	2.44	**−2.00**	**2.04**

Table 4: Unlearning performance on Qwen3.5-4B using TOFU-World Facts. Lower is better for M1–M2, M3 is best when closer to zero, and higher is better for M4.

Method	M1↓	M2↓	M3→ 0	M4↑
ReLearn	4.92	4.96	4.72	2.18
LUNAR	1.24	1.26	4.02	2.10
SURE+NPO	1.56	1.58	3.34	2.02
FOM-UL	**1.18**	**1.20**	**1.86**	**2.88**

Table 5: Efficiency comparison on TOFU-World Facts with Llama-2. FOM-UL achieves a practical balance between trainable parameter size, GPU memory, and runtime while avoiding full-model updates.

Method	Params.	GPU Mem.	Time
GA_{GDR}	7B	16 GB	4 hrs
NPO_{GDR}	7B	8 GB	3.2 hrs
KLD	7B	32 GB	4 hrs
SURE + NPO	**1.7M**	8 GB	**15 min**
ReLearn	2B	16 GB	35 min
LUNAR	1.75M	8 GB	20 min
FOM-UL	7M	**8 GB**	20 min

the same TOFU-World Facts protocol. As shown in Table 4, FOM-UL achieves the lowest residual memorization and privacy deviation while preserving substantially higher retain set utility. This result suggests that the proposed layer-selective forget-retain localization continues to provide a strong forgetting-utility trade-off on a newer and larger LLM.

4.4 Runtime Performance

Table 5 highlights the computational efficiency of different unlearning methods. Full-model approaches such as GA and KLD incur high memory and runtime costs due to updating all parameters, making them less practical for large-scale deployment. In contrast, parameter-efficient methods significantly reduce resource requirements. Notably, FOM-UL achieves competitive unlearning performance while updating only a small subset of parameters, requiring the lowest GPU memory (6–8 GB) and short runtimes (10–30 minutes). Compared to other efficient baselines such as SURE+NPO, ReLearn, and LUNAR, FOM-UL offers a more favorable balance between parameter count, memory usage, and execution time, demonstrating its practicality for scalable and resource-constrained unlearning settings. Details regarding parameter count in Appendix C, and reproducibility and hyperparameter configurations are provided in Appendix K.

4.5 Ablation Study

The ablation results in table-6 show that using only a single component of FOM-UL leads to suboptimal trade-offs between forgetting and utility. While `FOM-UL_forget_only` achieves low memorization, it suffers from high privacy leakage, and `FOM-UL_retain_only` preserves utility at the cost of weaker forgetting. `FOM-UL_mismatch_only` improves utility but introduces instability in privacy behavior. In contrast, **FOM-UL_full** consistently balances effective forgetting with stable privacy and utility, confirming the necessity of jointly optimizing all FOM-UL components. A comprehensive sensitivity analysis is provided in Appendix G.

Table 6: Ablation study of FOM-UL components on TOFU-World Facts with Llama-3.2-1B.

Variant	M1↓	M2↓	M3→ 0	M4↑
Forget only	1.56	1.56	8.31	2.83
Retain only	2.28	2.28	7.30	**4.62**
Mismatch only	1.56	1.56	−6.44	4.66
FOM-UL full	**1.24**	**1.24**	**2.04**	2.90

5 Discussion

Figure 5 results show that high-$\mathrm{Sig}(\ell)$ layer selection is more effective than early-only or over-expanded updates, and that FOM-UL also yields the lowest attack leakage rate under jailbreak-style recovery prompts. Full normalization and aggregation details are provided in Appendix C.

Overall, **FOM-UL** consistently reduces residual targeted knowledge with limited utility loss, and remains robust under aggressive post-training quantization, where many conventional unlearning methods exhibit severe reversals due to discretizations artifacts.

Extended Evaluation on Diverse LLM Architectures. We further validate generality across LLaMA-2, LLaMA-3, GPT-2, and Gemma-3 on the Tofu World Facts benchmark. Across architectures, FOM-UL reduces residual memorization and privacy leakage while maintaining strong retained utility. Under 4-bit quantization on Llama-3.2-1B, FOM-UL achieves M1=2.44 while retaining M4=2.04, demonstrating a favorable forgetting–utility trade-off under aggressive quantization.

FOM-UL's efficiency stems from: (i) **selective updates** restricted to a small set of responsible layers, (ii) **faster optimization** due to fewer trainable parameters per step, and (iii) **lower memory footprint**, enabling larger batches and reduced checkpointing. These gains make FOM-UL practical at scale and competitive with quantization-centric approaches in both completeness and utility (Zhang et al., 2024b), aligning with real-world compliance needs under evolving privacy regulations.

Additional implementation details and variabil-

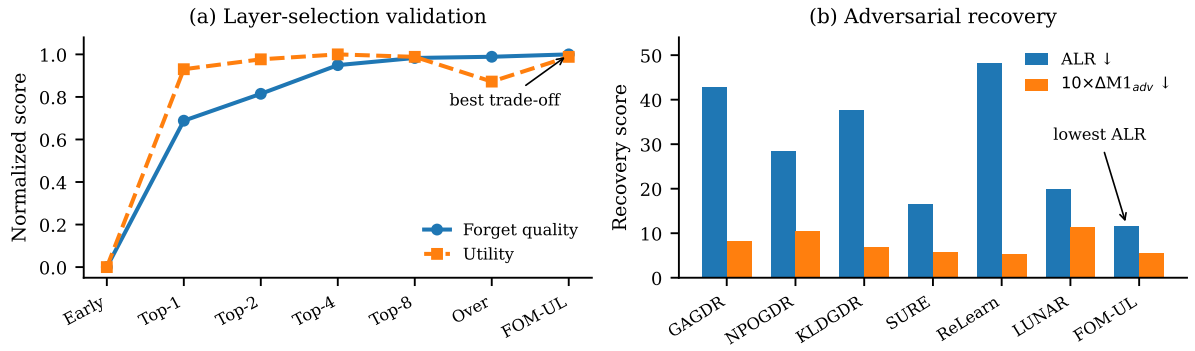


Figure 5: Layer-selection and robustness validation. FOM-UL achieves the best forgetting-utility trade-off across layer choices and the lowest adversarial recovery among evaluated baselines.

ity analyses are provided in the Appendix D. We also include ablations probing quantization-induced recovery by varying the update scope (top k layers through full-model), learning rates, and regularization strength. In contrast to diffuse global updates, FOM-UL concentrates stronger edits on attribution-identified layers, yielding robust forgetting with limited collateral damage (Zhang et al., 2024b).

6 Conclusion

We introduced **FOM-UL**, a layer-selective framework for approximate LLM unlearning. FOM-UL identifies layers with high forget-set influence and low retain-set sensitivity, then restricts updates to this small subset rather than modifying the full model. This design improves the forgetting-utility trade-off while reducing unnecessary parameter changes and computational cost.

Across TOFU, KnowUnDo, and MUSE-style evaluations, FOM-UL reduces residual memorization compared with strong unlearning baselines while preserving retain-set utility. Its concentrated updates also improve robustness under adversarial recovery prompts and low-bit post-training quantization, where diffuse unlearning updates can be weakened or erased. These results support layer-level forget-retain localization as a practical direction for efficient and deployment-aware unlearning, while leaving formal guarantees of complete erasure to future work.

Limitations

While FOM-UL improves targeted forgetting and quantization robustness by concentrating updates on a small subset of influential layers, its effectiveness depends on the reliability of layer-attribution signals, which can vary across prompts, domains, and evaluation setups. Moreover, FOM-UL is not a formal guarantee of erasure: highly entangled or redundantly encoded knowledge may require expanding the updated layer set or additional iterations, which can increase compute and introduce utility trade-offs under more adversarial or distribution-shifted settings. We discuss **ethical considerations**, safeguards, intended use, and risk mitigation in Appendix J.

AI Use Statement

Generative AI tools, including free-access GPT-based models, were used only for limited grammar and language refinement. All research ideas, methodology, experiments, analysis, and conclusions were developed and verified by the authors.

Acknowledgment

This material is based upon work co-supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Contract No. DE-AC05-00OR22725. This manuscript has been co-authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doepublic-access-plan).

References

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

Jiaao Chen and Diyi Yang. 2023. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*.

E Council and 1 others. 2022. General data protection regulation. *Official Journal of the European*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35:30318–30332.

Jiahui Geng, Qing Li, Herbert Woisetschlaeger, Zongxiong Chen, Fengyu Cai, Yuxia Wang, Preslav Nakov, Hans-Arno Jacobsen, and Fakhri Karray. 2025. A comprehensive survey of machine unlearning techniques for large language models. *arXiv preprint arXiv:2503.01854*.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2022. A survey of quantization methods for efficient neural network inference. In *Low-power computer vision*, pages 291–326. Chapman and Hall/CRC.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 383–398. Springer.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36:76033–76060.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

James Y Huang, Wenxuan Zhou, Fei Wang, Fred Morstatter, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2024. Offset unlearning for large language models. *arXiv preprint arXiv:2404.11045*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2022. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*.

Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Tianyi Alex Qiu, Juntao Dai, and Yaodong Yang. 2024. Aligner: Efficient alignment by learning to correct. *Advances in Neural Information Processing Systems*, 37:90853–90890.

Moxin Li, Yong Zhao, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See Kiong Ng, Tat-Seng Chua, and Yang Deng. 2025. Knowledge boundary of large language models: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5131–5157.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100.

Chris Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. 2024a. Large language model unlearning via embedding-corrupted prompts. *Advances in Neural Information Processing Systems*, 37:118198–118266.

Ollie Liu, Deqing Fu, Dani Yogatama, and Willie Neiswanger. 2024b. Dellma: Decision making under uncertainty with large language models. *arXiv preprint arXiv:2402.02392*.

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

Yezi Liu, Hanning Chen, Wenjun Huang, Yang Ni, and Mohsen Imani. 2025b. Lune: Efficient llm unlearning via lora fine-tuning with negative examples. *arXiv preprint arXiv:2512.07375*.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

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

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

Ravi Ranjan, Utkarsh Grover, Xiaomin Lin, and Agoritsa Polyzou. 2026. Razor: Ratio-aware layer editing for targeted unlearning in vision transformers and diffusion models. *arXiv preprint arXiv:2603.14819*.

William F Shen, Xinchi Qiu, Meghdad Kurmanji, Alex Iacob, Lorenzo Sani, Yihong Chen, Nicola Cancedda, and Nicholas D Lane. 2025. Llm unlearning via neural activation redirection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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 *International Conference on Learning Representations*, volume 2024, pages 51826–51843.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah Smith, and Chiyuan Zhang. 2025. Muse: Machine unlearning six-way evaluation for language models. In *International Conference on Learning Representations*, volume 2025, pages 27797–27818.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Qwen Team. 2026. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*.

Bozhong Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Mengru Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. 2024. To forget or not? towards practical knowledge unlearning for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1524–1537.

Bo Wang, Weiyi He, Shenglai Zeng, Zhen Xiang, Yue Xing, Jiliang Tang, and Pengfei He. 2025a. Unveiling privacy risks in llm agent memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25241–25260.

Changsheng Wang, Yihua Zhang, Dennis Wei, Jinghan Jia, Pin-Yu Chen, and Sijia Liu. 2025b. Llm unlearning on noisy forget sets: A study of incomplete, rewritten, and watermarked data. In *Proceedings of the 18th ACM Workshop on Artificial Intelligence and Security*, pages 136–145.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does llm safety training fail? *Advances in neural information processing systems*, 36:80079–80110.

Abudukelimu Wuerkaixi, Qizhou Wang, Sen Cui, Wutong Xu, Bo Han, Gang Niu, Masashi Sugiyama, and Changshui Zhang. 2025. Adaptive localization of knowledge negation for continual llm unlearning. In *Forty-second International Conference on Machine Learning*.

Derong Xu, Ziheng Zhang, Zhihong Zhu, Zhenxi Lin, Qidong Liu, Xian Wu, Tong Xu, Wanyu Wang, Yuyang Ye, Xiangyu Zhao, and 1 others. 2024. Editing factual knowledge and explanatory ability of medical large language models. In *Proceedings of the 33rd ACM international conference on information and knowledge management*, pages 2660–2670.

Haoming Xu, Ningyuan Zhao, Liming Yang, Sendong Zhao, Shumin Deng, Mengru Wang, Bryan Hooi, Nay Oo, Huajun Chen, and Ningyu Zhang. 2025. Relearn: Unlearning via learning for large language models. *arXiv preprint arXiv:2502.11190*.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024a. Machine unlearning of pre-trained large language models. *arXiv preprint arXiv:2402.15159*.

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

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024a. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. 2024b. Catastrophic failure of llm unlearning via quantization. *arXiv preprint arXiv:2410.16454*.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. 2024c. Does your llm truly unlearn? an embarrassingly simple approach to recover unlearned knowledge. *arXiv e-prints*, pages arXiv–2410.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Yang Zhao, Hongyang Du, Yijing Lin, Keyi Xiang, Dusit Niyato, and H Vincent Poor. 2025. A survey on continuous unlearning in generative ai: Approaches and trade-offs. *IEEE Intelligent Systems*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

Appendix

A Pseudocode

B Experimental Settings

B.1 Models and Initialization

We evaluate FOM-UL on diverse transformer LLMs to test model-agnostic behavior, using pretrained checkpoints from Meta Llama (Llama-2 7B, Llama-3.2 1B), GPT-2, and Gemma-3 1B. Each run starts from the base parameters θ_0, and produces an unlearned checkpoint $\theta^\star$ by updating only a small set of layers S selected via the forget-to-retain gradient significance score, while all other layers remain frozen. We also report robustness under post-training quantization by evaluating $\theta^\star$ in FP32, 8-bit, and 4-bit formats.

B.2 Datasets and Splits

We follow standard unlearning protocols with disjoint **Forget** and **Retain** splits, and benchmark on TOFU (world-facts QA), KnowUnDo, and MUSE (Shi et al., 2025) (BOOKS/NEWS). For BOOKS, D_{forget} contains copyrighted *Harry Potter* text and D_{retain} includes FanWiki (plus general-domain text) to preserve non-verbatim knowledge while removing memorization. For NEWS, we additionally use a **holdout** split reserved for privacy/leakage evaluation and never used for updates.

B.3 Training Configuration

FOM-UL performs masked updates with three objectives: (i) gradient ascent on L_{forget}, (ii) gradient ascent on L_{mismatch} to repel original outputs on forget prompts, and (iii) gradient descent on L_{retain} to preserve utility. We use AdamW with consistent hyperparameters across methods; unless stated otherwise, we set batch size $B = 16$, learning rate 1×10^{-5}, and run unlearning for 5 epochs. Random seeds are fixed (e.g., 42) for reproducibility.

B.4 Implementation Details

All methods are implemented in PyTorch using HuggingFace `transformers`. Quantization is performed with `bitsandbytes` to produce FP32/8-bit/4-bit variants for robustness checks. Experiments run on NVIDIA A100 GPUs (40 GB), and we release environment details and fixed splits to support reproducibility.

C Evaluation Metrics

We evaluate FOM-UL using four standard metrics (M1-M4) that jointly quantify (i) how well the model forgets the designated forget set and (ii) how well it preserves utility on the retain set, following the *SURE* evaluation protocol.

Notation. Let f denote an LLM. Let $\mathcal{D}_{\text{forget}}$ be the forget set, $\mathcal{D}_{\text{retain}}$ the retain set, and $\mathcal{D}_{\text{holdout}}$ a disjoint holdout set used for privacy auditing. For ROUGE-based metrics, $\mathrm{ROUGE}(\cdot,\cdot)$ measures similarity between the model output and a reference text.

M1: Verbatim Memorization (VerMem) on $\mathcal{D}_{\text{forget}}$ (lower is better). Given a forget document x tokenized into a prefix $x_{1:\ell}$ and ground-truth continuation $x_{\ell+1:}$, we compute:

$$\mathrm{M1}(f) \;=\; \mathbb{E}_{x\sim\mathcal{D}_{\text{forget}}}\Big[\mathrm{ROUGE}\big(f(x_{1:\ell}),\, x_{\ell+1:}\big)\Big]. \tag{12}$$

This captures *verbatim* reproduction of forgotten content; effective unlearning drives M1 down.

M2: Knowledge Memorization (KnowMem) on $\mathcal{D}_{\text{forget}}$ (lower is better). Using knowledge-oriented QA pairs (q, a) derived from the forget set, we measure whether the model still answers with forgotten knowledge:

$$\mathrm{M2}(f) \;=\; \mathbb{E}_{(q,a)\sim\mathcal{D}_{\text{forget}}}\Big[\mathrm{ROUGE}\big(f(q),\, a\big)\Big]. \tag{13}$$

Lower values indicate better removal of generalized (non-verbatim) forgotten knowledge.

M3: Privacy Leakage (PrivLeak) via membership inference (closer to 0 is better). We quantify privacy risk using a membership inference attack based on the Min-$K\%$ criterion, which produces an AUC-ROC score $\mathrm{AUC}(f)$ by distinguishing samples from $\mathcal{D}_{\text{forget}}$ vs. $\mathcal{D}_{\text{holdout}}$. We then compare against a retrained baseline f_{retrain} (trained without $\mathcal{D}_{\text{forget}}$) and define:

$$\mathrm{M3}(f) \;=\; \frac{\mathrm{AUC}(f) - \mathrm{AUC}(f_{\text{retrain}})}{\mathrm{AUC}(f)}. \tag{14}$$

An ideal unlearned model matches the retrained privacy behavior, yielding $\mathrm{M3} \approx 0$; large deviations indicate elevated privacy leakage.

Interpretation of negative M3. From above metric definition, $M3(f) = \big(\mathrm{AUC}(f) - \mathrm{AUC}(f_{\text{retrain}})\big)/\mathrm{AUC}(f)$, so $M3 < 0$ occurs whenever $\mathrm{AUC}(f) < \mathrm{AUC}(f_{\text{retrain}})$. Such

Algorithm 1 FOM-UL: Forgetting Only What Matters via Unlearning Layers

Require: Pretrained parameters $\theta = \{\theta^{(1)}, \ldots, \theta^{(L)}\}$; forget set D_f; retain set D_r; losses $\mathcal{L}_f$, $\mathcal{L}_m$, $\mathcal{L}_r$; learning rates η_f, η_m, η_r; steps T; initial layer budget k; tolerance $\epsilon > 0$.
Ensure: Unlearned parameters θ_u.

Stage 1: Forget-retain layer attribution

for $\ell = 1, \ldots, L$ **do**
  Sample mini-batches $B_f \subset D_f$ and $B_r \subset D_r$.
  $I_f(\ell) \leftarrow \|\nabla_{\theta^{(\ell)}} \mathcal{L}_f(\theta; B_f)\|_2$
  $I_r(\ell) \leftarrow \|\nabla_{\theta^{(\ell)}} \mathcal{L}_r(\theta; B_r)\|_2$
  $\mathrm{Sig}(\ell) \leftarrow I_f(\ell)/(I_r(\ell) + \epsilon)$
end for
Rank layers by $\mathrm{Sig}(\ell)$ in descending order and denote the ordering by π.
Initialize selected layer set $S \leftarrow \{\pi_1, \ldots, \pi_k\}$ and $\theta_0 \leftarrow \theta$.

Stage 2: Selective layer-wise unlearning

for $t = 0, \ldots, T-1$ **do**
  Sample mini-batches $B_f \subset D_f$ and $B_r \subset D_r$.
  for $\ell \in S$ **do**
    $g_f \leftarrow \nabla_{\theta^{(\ell)}} \mathcal{L}_f(\theta_t; B_f)$
    $g_m \leftarrow \nabla_{\theta^{(\ell)}} \mathcal{L}_m(\theta_t; B_f, B_r)$
    $g_r \leftarrow \nabla_{\theta^{(\ell)}} \mathcal{L}_r(\theta_t; B_r)$
    $\theta_{t+1}^{(\ell)} \leftarrow \theta_t^{(\ell)} + \eta_f g_f + \eta_m g_m - \eta_r g_r$
  end for
  for $\ell \notin S$ **do**
    $\theta_{t+1}^{(\ell)} \leftarrow \theta_t^{(\ell)}$ ▷ freeze non-selected layers
  end for
end for

Stage 3: Metric check and iterative expansion

Evaluate θ_T using VerMem↓, KnowMem↓, PrivLeak→ 0, and Utility↑.
if forgetting criteria are not satisfied **then**
  Expand $S \leftarrow S \cup \{\arg\max_{\ell \notin S} \mathrm{Sig}(\ell)\}$.
  Repeat Stage 2 with the expanded layer set.
end if
return $\theta_u \leftarrow \theta_T$.

strongly negative values (e.g., -2.00 in Table 3) do *not* mean "better privacy than zero"; rather, they indicate a large deviation from the retrained privacy behavior, typically corresponding to *over-unlearning* in which the forget examples become atypically high-loss relative to holdout, flipping or amplifying the attack signal. Because our normalization divides by $\mathrm{AUC}(f)$, the magnitude can exceed 1 when $\mathrm{AUC}(f)$ is small; for instance, if $\mathrm{AUC}(f) = 0.20$ and $\mathrm{AUC}(f_{\text{retrain}}) = 0.60$, then $M3 = (0.20 - 0.60)/0.20 = -2.0$. Therefore, we interpret $M3$ by distance to zero: $|M3|$ large (positive or negative) implies unstable privacy behavior, while $M3 \approx 0$ indicates the closest match to retraining.

M4: Utility Preservation on $\mathcal{D}_{\text{retain}}$ (higher is better). We measure retained utility using the same KnowMem-style QA evaluation on $\mathcal{D}_{\text{retain}}$:

$$\mathrm{M4}(f) \;=\; \mathbb{E}_{(q,a)\sim\mathcal{D}_{\text{retain}}}\Big[\mathrm{ROUGE}\big(f(q),\, a\big)\Big]. \tag{15}$$

Higher $\mathrm{M4}$ indicates better preservation of benign knowledge and task utility after unlearning.

Overall objective. FOM-UL aims to achieve low $\mathrm{M1}$/$\mathrm{M2}$ (forgetting), $\mathrm{M3} \approx 0$ (privacy parity with retraining), and high $\mathrm{M4}$ (utility), including under post-training quantization stress tests highlighted in our study.

Runtime Performance parameter counts and efficiency. In Table 4, the *Parameters* column reports the number of *trainable* (i.e., unfrozen) parameters that each method actually updates during unlearning, not the total backbone size of Llama-2 7B. Accordingly, full-model baselines (e.g., GA/KLD/NPO-GDR) update the entire parameter space, whereas parameter-efficient or masked-update methods (SURE+NPO, LUNAR, and FOM-UL) optimize only a small, selected subset, yielding 1.7M, 1.75M, and 7M trainable parameters, respectively. For FOM-UL, the backbone remains frozen and updates are confined to the selected layer set S (and the specific trainable submodules within those layers), so the trainable-parameter count scales with the layer budget and the chosen update parameterization.

Clarification (Fig. 4 M.U. and F.Q.). For Fig. 4,

we define *Normalized Utility (M.U.)* as the min-max normalization of $M4$ across methods and steps, $\mathrm{M.U.}(t) = \frac{M4(t)-\min M4}{\max M4-\min M4}$, so higher is better. To ensure *Forget Quality (F.Q.)* increases when forgetting improves (since $M1$-$M3$ are $\downarrow$ metrics), we first invert and normalize each metric as $\tilde{M}_i(t) = 1 - \frac{M_i(t)-\min M_i}{\max M_i-\min M_i}$ for $i \in \{1,2,3\}$, and then aggregate by a weighted mean $\mathrm{F.Q.}(t) = \frac{1}{3}\sum_{i=1}^{3} \tilde{M}_i(t)$ (weights set uniformly unless stated otherwise).

Update parameterization. FOM-UL is layer-selective at the selection level but submodule-selective at the implementation level. After selecting layer set S, we enable gradients only for the chosen trainable submodules inside those layers, e.g., attention projection and/or MLP projection matrices depending on the backbone implementation. All layers $\ell \notin S$ and all disabled submodules inside $\ell \in S$ remain frozen. Thus, the reported trainable-parameter count is

$$N_{\text{train}} = \sum_{\ell \in S} \sum_{m \in \mathcal{M}_\ell} |\theta^{(\ell,m)}|, \tag{16}$$

where $\mathcal{M}_\ell$ is the set of enabled submodules in layer ℓ. This count measures parameters actually updated during unlearning, not the total number of parameters contained in the selected transformer layers.

D Detailed Methodology

D.1 Selecting and Masking Important Layers

In FOM-UL, Δ_ℓ (ablation) is used primarily as a diagnostic to motivate layer localization, while $\mathrm{Sig}(\ell)$ is the actual algorithmic criterion used to select and expand the update set.

Layer selection via forget-retain significance. Let f_θ denote a transformer-based LLM with L layers and parameters $\theta = \{\theta^{(1)}, \ldots, \theta^{(L)}\}$ grouped by layer. FOM-UL selects a small subset of layers whose parameters are most responsive to the forgetting objective while minimally affecting retained utility. Given a forget mini-batch $B_f \subset \mathcal{D}_{\text{forget}}$ and a retain mini-batch $B_r \subset \mathcal{D}_{\text{retain}}$, define the corresponding losses $\mathcal{L}_{\text{forget}}(\theta; B_f)$ and $\mathcal{L}_{\text{retain}}(\theta; B_r)$. For each layer ℓ, we compute per-layer gradient magnitudes:

$$\begin{aligned} I(\ell) &= \left\| \nabla_{\theta^{(\ell)}} \mathcal{L}_{\text{forget}}(\theta; B_f) \right\|_2, \\ I_r(\ell) &= \left\| \nabla_{\theta^{(\ell)}} \mathcal{L}_{\text{retain}}(\theta; B_r) \right\|_2, \end{aligned} \tag{17}$$

and the forget-retain *significance ratio*

$$\mathrm{Sig}(\ell) = \frac{I(\ell)}{I_r(\ell) + \varepsilon}, \tag{18}$$

where $\varepsilon > 0$ ensures numerical stability. Intuitively, high $\mathrm{Sig}(\ell)$ indicates strong forgetting leverage with limited interference on retained knowledge. We select the initial update set using thresholding (or equivalently top-k ranking):

$$S = \{\ell \in \{1, \ldots, L\} : \mathrm{Sig}(\ell) > \tau\}, \tag{19}$$

where τ controls the layer budget.

Binary layer mask. We define a layer mask $m_\ell \in \{0, 1\}$ as

$$m_\ell = \begin{cases} 1, & \ell \in S, \\ 0, & \text{otherwise}, \end{cases} \tag{20}$$

so that only layers in S receive gradient updates and all other layers remain frozen.

When $\mathrm{Sig}(\ell)$ is computed. For reproducibility, we compute $\mathrm{Sig}(\ell)$ *once* at initialization using the base checkpoint θ_0 and a fixed mini-batch (or small fixed set) sampled from $\mathcal{D}_{\text{forget}}$ and $\mathcal{D}_{\text{retain}}$, and we keep this ranking *static* during unlearning. During iterative expansion, we do *not* recompute gradients each epoch; instead, we expand S by adding the next highest-ranked layers under this fixed $\mathrm{Sig}(\ell)$ ordering until the stopping criterion is met. (Optionally, one may recompute $\mathrm{Sig}(\ell)$ at expansion boundaries as an ablation, but our main results use the static initialization for stability and determinism.)

D.2 Selective Layer Updates with Forget, Mismatch, and Retain Losses

Forgetting loss $\mathcal{L}_{\text{forget}}$. FOM-UL enforces targeted forgetting by *increasing* the model's loss on forget-set continuations so that memorized responses become unlikely. For a forget example $(x, y) \in \mathcal{D}_{\text{forget}}$ with prompt x and reference continuation $y = (y_1, \ldots, y_m)$, we use the standard negative log-likelihood (NLL):

$$\begin{aligned} \mathcal{L}_{\text{forget}}(\theta) &= \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{forget}}} \\ &\Big[-\sum_{i=1}^{m} \log p_\theta(y_i \mid x, y_{<i}) \Big], \end{aligned} \tag{21}$$

and perform *gradient ascent* on $\mathcal{L}_{\text{forget}}$ (cf. Eq. 9), which directly reduces the likelihood of generating the memorized forget content (Yao et al., 2024b,a).

Mismatch loss $\mathcal{L}_{\text{mismatch}}$. A core goal of FOM-UL is to actively *diverge* from the original model's behavior on the forget set, rather than merely reducing confidence in a single target token. To operationalize this, we define a *distribution-level* mismatch objective that pushes the unlearned model

away from the original model's output distribution on forget prompts. Let f_{θ_0} denote the original (pre-unlearning) model and f_θ the current (unlearned) model. For a forget prompt $x \in \mathcal{D}_{\text{forget}}$, let $z_\theta(x) \in \mathbb{R}^{|\mathcal{V}|}$ be the next-token logits and define temperature-scaled predictive distributions

$$\begin{aligned} p_\theta(\cdot \mid x) &= \operatorname{softmax}\left(\frac{z_\theta(x)}{T}\right), \\ p_{\theta_0}(\cdot \mid x) &= \operatorname{softmax}\left(\frac{z_{\theta_0}(x)}{T}\right), \end{aligned} \tag{22}$$

where $T \geq 1$ controls how strongly the loss emphasizes high-probability tokens. We then set

$$\begin{aligned} \mathcal{L}_{\text{mismatch}}(\theta) &= \mathbb{E}_{x \sim \mathcal{D}_{\text{forget}}} \\ &\Big[D_{\text{KL}}\big(p_{\theta_0}(\cdot \mid x) \,\|\, p_\theta(\cdot \mid x)\big) \Big], \end{aligned} \tag{23}$$

and *maximize* $\mathcal{L}_{\text{mismatch}}$ during unlearning (cf. Eq. 9), which encourages f_θ to move its outputs away from the original model on the forget set. This objective is inspired by prior unlearning/alignment formulations that use KL-based output constraints to control behavior shifts (typically on retain data), here repurposed as an explicit *divergence* signal on the forget distribution (Yao et al., 2024b,a). In practice, $\mathcal{L}_{\text{mismatch}}$ helps prevent *incomplete forgetting* where the model's distribution remains close to p_{θ_0} despite reduced confidence in a specific answer, and it complements $\mathcal{L}_{\text{forget}}$ to mitigate quantization-induced recovery by enforcing a broader, distributional departure from the pre-unlearning behavior (Zhang et al., 2024b).

Retention loss $\mathcal{L}_{\text{retain}}$. To preserve utility on non-targeted behaviors, FOM-UL simultaneously minimizes a retain objective on benign data $\mathcal{D}_{\text{retain}}$. For a retain example $(x, y) \in \mathcal{D}_{\text{retain}}$, we again use the NLL:

$$\begin{aligned} \mathcal{L}_{\text{retain}}(\theta) &= \mathbb{E}_{(x,y) \sim \mathcal{D}_{\text{retain}}} \\ &\Big[-\sum_{i=1}^{m} \log p_\theta(y_i \mid x, y_{<i}) \Big], \end{aligned} \tag{24}$$

and apply *gradient descent* on $\mathcal{L}_{\text{retain}}$ to maintain general knowledge and task performance while unlearning is confined to the selected layers (Liu et al., 2024a; Yao et al., 2024a). When combined with $\mathcal{L}_{\text{forget}}$ and the divergence-based $\mathcal{L}_{\text{mismatch}}$, this retain term stabilizes optimization and mitigates collateral degradation.

FOM-UL performs targeted optimization only over $\{\theta^{(\ell)} : \ell \in S\}$ using three objectives: (i) a forgetting loss $\mathcal{L}_{\text{forget}}$ to suppress targeted content, (ii) a mismatch loss $\mathcal{L}_{\text{mismatch}}$ to diverge from the original model behavior on the forget set, and (iii) a retain loss $\mathcal{L}_{\text{retain}}$ to preserve general utility. The masked update for each layer ℓ at iteration t is:

$$\begin{aligned} \theta_{t+1}^{(\ell)} = \theta_t^{(\ell)} + m_\ell \Big(& \eta_F \nabla_{\theta^{(\ell)}} \mathcal{L}_{\text{forget}} \\ & + \eta_M \nabla_{\theta^{(\ell)}} \mathcal{L}_{\text{mismatch}} \\ & - \eta_R \nabla_{\theta^{(\ell)}} \mathcal{L}_{\text{retain}} \Big), \end{aligned} \tag{25}$$

where η_F, η_M, η_R are step sizes for the respective terms. Layers not selected for unlearning remain unchanged:

$$\theta_{t+1}^{(\ell)} = \theta_t^{(\ell)} \quad \forall \ell \notin S. \tag{26}$$

D.3 Iterative Expansion and Stopping Criteria

FOM-UL applies an iterative schedule that expands the update set only when forgetting criteria are unmet, limiting unnecessary intervention. After each unlearning round, we evaluate a forgetting criterion (e.g., VerMem below a threshold) and stop when it is satisfied. Otherwise, we expand the layer set by adding the next most significant layer under $\text{Sig}(\ell)$:

$$S \leftarrow S \cup \left\{ \arg\max_{\ell \notin S} \text{Sig}(\ell) \right\}, \tag{27}$$

and repeat the masked updates. This progressive expansion prevents overly aggressive early updates and reduces the risk of collateral utility loss.

Clarification (Initialization of iterative expansion). To remove ambiguity, we define the initialization as a *single, deterministic rule* based on $\text{Sig}(\ell)$: we first compute $\text{Sig}(\ell)$ for all layers and form the candidate set $S_\tau = \{\ell : \text{Sig}(\ell) \geq \tau\}$. If $|S_\tau| > k$, we take the top-k layers within S_τ by descending $\text{Sig}(\ell)$; if $|S_\tau| \leq k$, we simply set $S = S_\tau$. Equivalently, the threshold τ is the *primary filter* (ensuring a minimum forget-to-retain ratio), and k is an optional *budget cap* that prevents overly large initial updates. Iterative expansion then proceeds by adding one (or a small batch of) next-highest $\text{Sig}(\ell)$ layers from $\{\ell \notin S\}$ until the stopping criterion is met.

D.4 Robustness to Quantization-Induced Relearning

Post-training quantization maps full-precision parameters to a discrete set of representable values. When an unlearning method produces only

small, diffuse weight changes, quantization can erase these differences, yielding quantized models that are nearly indistinguishable from the original. FOM-UL mitigates this failure mode by concentrating updates within a small set of high-$\mathrm{Sig}(\ell)$ layers, producing more salient (layer-localized) parameter shifts while leaving the majority of layers unchanged. As a result, the intended forgetting signal is less likely to be collapsed by discretization, improving robustness against quantization-induced recovery compared to global-update baselines.

D.5 Empirical Quantization-Bin Crossing Analysis

An unlearning update on coordinate j is removed by quantization exactly when

$$\begin{aligned} & Q_{\Delta_j}(\theta'_j) = Q_{\Delta_j}(\theta_j) \\ \iff & \mathrm{Round}\left(\frac{\theta'_j}{\Delta_j}\right) = \mathrm{Round}\left(\frac{\theta_j}{\Delta_j}\right). \end{aligned} \tag{28}$$

Thus, quantization robustness is measured directly by the fraction of edited coordinates whose quantization indices change.

For each updated layer ℓ, let Δ_ℓ denote the effective quantization step size and $\delta\theta^{(\ell)} = \theta_u^{(\ell)} - \theta_0^{(\ell)}$ denote the unlearning update. We report the bin-crossing fraction

$$\mathrm{BCF}_\ell = \frac{1}{|\theta^{(\ell)}|} \sum_j \mathbb{I}\left[|\delta\theta_j^{(\ell)}| \geq \Delta_\ell/2\right], \tag{29}$$

and the normalized update ratio

$$\rho_\ell = \mathrm{median}_j \left(\frac{|\delta\theta_j^{(\ell)}|}{\Delta_\ell/2 + \epsilon}\right). \tag{30}$$

Higher BCF indicates that more unlearning edits survive post-training quantization.

Bin-Change Fraction. To directly measure whether unlearning updates survive quantization, we define the *Bin-Change Fraction* (BCF) over the set of edited parameters $\mathcal{E}$ as

$$\mathrm{BCF} = \frac{1}{|\mathcal{E}|} \sum_{j \in \mathcal{E}} \mathbf{1}\left[Q_{\Delta_j}(\theta'_j) \neq Q_{\Delta_j}(\theta_j)\right], \tag{31}$$

where θ_j and θ'_j denote the pre- and post-unlearning parameters, respectively, and Q_{Δ_j} denotes the corresponding quantization operator. A larger BCF indicates that a greater fraction of the unlearning-induced parameter changes remain distinguishable after quantization.

Table 7: **Quantization-bin crossing analysis on Llama-3.2-1B.** BCF reports the fraction of updated weights under 4-bit quantization. FOM-UL produces more quantization-surviving edits in selected layers than diffuse global updates.

Method	Updated scope	Mean BCF↑	ρ ratio↑
GAGDR	full model	0.7%	0.31
NPOGDR	full model	0.9%	0.36
SURE+NPO	sparse modules	2.8%	0.74
LUNAR	activation-targeted	2.1%	0.68
FOM-UL	selected layers	**5.6%**	**1.18**

Table 7 provides mechanism-level evidence for the quantization robustness claim. Compared with global baselines, FOM-UL concentrates larger updates in selected layers, producing a higher fraction of weights that cross 4-bit quantization bin boundaries and are therefore less likely to be rounded back to the original quantized value.

E Lemma justification for FOM-UL

Let the forget and retain objectives be $\mathcal{L}_f(\theta)$ and $\mathcal{L}_r(\theta)$, and let $\theta = \{\theta^{(1)}, \ldots, \theta^{(L)}\}$ denote parameters grouped by transformer layer. Define layer-wise gradients $g_f^{(\ell)} := \nabla_{\theta^{(\ell)}} \mathcal{L}_f(\theta)$ and $g_r^{(\ell)} := \nabla_{\theta^{(\ell)}} \mathcal{L}_r(\theta)$.

Lemma 1 (Greedy layer ranking under a retain-stability constraint)**.** *Consider one unlearning step that applies a layer-wise update $\Delta\theta = \{\Delta\theta^{(1)}, \ldots, \Delta\theta^{(L)}\}$, but is restricted to at most k layers (i.e., $\Delta\theta^{(\ell)} \neq 0$ only if $\ell \in S$, $|S| = k$). Assume a first-order approximation and impose a retain-stability constraint $\langle g_r^{(\ell)}, \Delta\theta^{(\ell)} \rangle \approx 0$ in sign (or bounded magnitude) so that retain utility is not degraded. Then, among candidate layers, the layers that maximize the achievable forget effect per unit retain sensitivity are those with the largest score*

$$\mathrm{Sig}(\ell) \;:=\; \frac{\|g_f^{(\ell)}\|}{\|g_r^{(\ell)}\| + \epsilon}, \tag{32}$$

for a small $\epsilon > 0$. Equivalently, selecting the top-k layers by $\mathrm{Sig}(\ell)$ is the greedy choice that prioritizes high forget responsiveness while being conservative on retain disruption.

Proof sketch. Using a first-order Taylor expansion

for a small step,

$$\Delta\mathcal{L}_f \;\approx\; \sum_{\ell=1}^{L} \left\langle g_f^{(\ell)}, \Delta\theta^{(\ell)} \right\rangle, \qquad \Delta\mathcal{L}_r \;\approx\; \sum_{\ell=1}^{L} \left\langle g_r^{(\ell)}, \Delta\theta^{(\ell)} \right\rangle. \tag{33}$$

For a given layer ℓ, the maximum attainable forget change from updating only that layer under a step-size budget $\|\Delta\theta^{(\ell)}\| \le \rho$ is bounded by Cauchy-Schwarz:

$$\left\langle g_f^{(\ell)}, \Delta\theta^{(\ell)} \right\rangle \;\le\; \|g_f^{(\ell)}\|\,\|\Delta\theta^{(\ell)}\| \;\le\; \rho\,\|g_f^{(\ell)}\|. \tag{34}$$

At the same time, the magnitude of the retain change contributed by layer ℓ is similarly bounded:

$$\left|\left\langle g_r^{(\ell)}, \Delta\theta^{(\ell)} \right\rangle\right| \;\le\; \|g_r^{(\ell)}\|\,\|\Delta\theta^{(\ell)}\| \;\le\; \rho\,\|g_r^{(\ell)}\|. \tag{35}$$

Thus, a natural "forget gain per retain sensitivity" proxy for layer ℓ is $\|g_f^{(\ell)}\|/(\|g_r^{(\ell)}\| + \epsilon)$, where ϵ stabilizes the ratio when $\|g_r^{(\ell)}\|$ is small. Selecting the top-k layers by this ratio maximizes the sum of these proxies under a k-sparsity constraint, which is exactly the FOM-UL ranking rule. □

Remark (what this proves and what it doesn't). Lemma 1 justifies FOM-UL's layer ranking as an *optimal greedy criterion* under (i) small-step / first-order behavior and (ii) a utility-preservation constraint expressed through retain gradients. It does not claim global optimality for deep, non-convex objectives; rather it gives a principled reason that the gradient-ratio score is the right layer selection signal.

Lemma 2 (Iterative expansion is a monotone relaxation)**.** *Let $\mathcal{L}_f(\theta)$ be the forget objective and $\mathcal{L}_r(\theta)$ be the retain objective. Fix a current parameter state θ and consider a* one-step *selective update $\Delta\theta$ that is allowed to modify only layers in a set $S \subseteq \{1, \dots, L\}$. Define the feasible update set*

$$\mathcal{U}(S) \;:=\; \Big\{\Delta\theta :\; \Delta\theta^{(\ell)} = 0\;\forall \ell \notin S, \;\; \|\Delta\theta^{(\ell)}\| \le \rho\;\forall\ell,\;\; |\Delta\mathcal{L}_r(\theta;\Delta\theta)| \le \delta\Big\}, \tag{36}$$

where $\Delta\mathcal{L}_r(\theta;\Delta\theta)$ denotes the first-order retain change $\Delta\mathcal{L}_r \approx \sum_\ell \langle \nabla_{\theta^{(\ell)}}\mathcal{L}_r(\theta), \Delta\theta^{(\ell)}\rangle$, ρ is a step-size budget, and δ is a tolerance for retain degradation. Let the best achievable forget decrease (first-order) under S be

$$V(S) \;:=\; \min_{\Delta\theta\in\mathcal{U}(S)} \Delta\mathcal{L}_f(\theta;\Delta\theta), \qquad \Delta\mathcal{L}_f(\theta;\Delta\theta) \approx \sum_\ell \langle \nabla_{\theta^{(\ell)}}\mathcal{L}_f(\theta), \Delta\theta^{(\ell)}\rangle. \tag{37}$$

If $S \subseteq S'$ (i.e., S' expands S with additional layers), then

$$V(S') \;\le\; V(S). \tag{38}$$

Equivalently, expanding the set of trainable layers cannot worsen the best attainable forgetting progress under the same retain-stability constraint. This provides a principled justification for FOM-UL's iterative expansion rule (e.g., $k \to k + k'$) when residual memorization remains above a threshold.

Proof. Because $S \subseteq S'$, any update $\Delta\theta$ that is feasible for S is also feasible for S': we can view it as an element of $\mathcal{U}(S')$ by simply setting $\Delta\theta^{(\ell)} = 0$ for all newly added layers $\ell \in S' \setminus S$. Hence $\mathcal{U}(S) \subseteq \mathcal{U}(S')$. Minimizing the same objective $\Delta\mathcal{L}_f$ over a superset of feasible points cannot yield a worse optimum, so $\min_{\Delta\theta\in\mathcal{U}(S')} \Delta\mathcal{L}_f \le \min_{\Delta\theta\in\mathcal{U}(S)} \Delta\mathcal{L}_f$, i.e., $V(S') \le V(S)$. □

Practical interpretation. If the current top-k selected layers S do not sufficiently reduce memorization while meeting the retain constraint, expanding S (adding the next-ranked k' layers) strictly relaxes the optimization problem. Therefore FOM-UL's iterative expansion is a safe strategy: it never removes previously feasible updates, and can only maintain or improve the best achievable forgetting progress subject to utility preservation.

Theorem 1 (Quantization Persistence for Layer-Selective Updates)**.** *Let f_θ be a pretrained model and let $f_{\theta'}$ be the unlearned model obtained by updating only layers in $S \subseteq \{1, \dots, L\}$ (all $\ell \notin S$ are frozen). Consider post-training uniform symmetric rounding quantization applied elementwise (or per-group) with step size $\Delta_\ell > 0$ for layer ℓ:*

$$Q_{\Delta_\ell}(w) \;=\; \Delta_\ell \cdot \mathrm{Round}\left(\frac{w}{\Delta_\ell}\right). \tag{39}$$

Define the layerwise update $\Delta\theta^{(\ell)} := \theta'^{(\ell)} - \theta^{(\ell)}$.

Table 8: **Adversarial/jailbreak robustness metrics.** We evaluate TOFU-World Facts on Llama-3.2-1B using adversarial prompt wrappers $\mathcal{A}$, including role-play, instruction-override, extraction-style, and suffix-based jailbreak prompts (Wei et al., 2023; Zou et al., 2023). $M1_{\text{adv}}$, $M2_{\text{adv}}$, and ALR are lower-is-better; $M3_{\text{adv}}$ is best when closest to zero; and $M4$ is higher-is-better.

Metric ID	Name	Definition and Interpretation	Goal
$M1_{\text{adv}}$	Adv. VerMem	$\mathbb{E}_{x\sim D_f,\, a\sim\mathcal{A}}\,[\text{ROUGE}(f(a(x_{1:\ell})), x_{\ell+1:})]$. Measures whether the model can reproduce verbatim forgotten continuations when the forget prefix is wrapped by a jailbreak prompt.	↓
$M2_{\text{adv}}$	Adv. KnowMem	$\mathbb{E}_{(q,y)\sim D_f,\, a\sim\mathcal{A}}\,[\text{ROUGE}(f(a(q)), y)]$. Measures recovery of forgotten QA-style knowledge under adversarial prompting, following TOFU-style knowledge memorization evaluation (Maini et al., 2024).	↓
$M3_{\text{adv}}$	Adv. PrivLeak	$\frac{\text{AUC}_{\text{adv}}(f) - \text{AUC}(f_{\text{retrain}})}{\text{AUC}_{\text{adv}}(f)}$. Here, $\text{AUC}_{\text{adv}}(f)$ is computed using Min-$K\%$ membership scores over jailbreak-wrapped forget and holdout prompts (Shi et al., 2024). Values closer to zero indicate privacy behavior closer to retraining.	$\to 0$
ALR	Attack Leakage Rate	$100 \cdot \mathbb{E}_{(q,y)\sim D_f,\, a\sim\mathcal{A}}\,[\mathbb{1}\{\text{ROUGE}(f(a(q)), y) > \tau_r\}]$. Reports the percentage of adversarial prompts that recover the forgotten answer above threshold τ_r. We set $\tau_r = 0.30$ following the compact ROUGE-style reporting scale.	↓
$M4$	Retain Utility	$\mathbb{E}_{(q,y)\sim D_r}\,[\text{ROUGE}(f(q), y)]$. Measured on clean retain prompts to verify that adversarial robustness is not achieved through destructive over-unlearning or general utility collapse.	↑

Proposition (1) (Quantization Persistence)**.** Let θ and θ' denote the parameters before and after unlearning, and let $\mathcal{E}$ be the set of edited coordinates. If

$$Q_{\Delta_j}(\theta'_j) = Q_{\Delta_j}(\theta_j), \qquad \forall j \in \mathcal{E}, \tag{40}$$

then the edited coordinates are indistinguishable from their pre-unlearning values under the quantizer Q. Consequently, the quantized model does not preserve these parameter-level unlearning changes.

This result does not imply that identical quantized parameters necessarily produce identical model behavior in every setting; rather, it characterizes when parameter updates introduced by unlearning are removed by the quantization operator.

Proof sketch. Under rounding quantization, a real value w is mapped to the nearest grid point with spacing Δ_ℓ; boundaries between adjacent quantization bins occur at half-steps. Therefore, changing w by at least $\Delta_\ell/2$ is sufficient to cross a bin boundary and change the quantization index, yielding $Q_{\Delta_\ell}(w+\delta) \neq Q_{\Delta_\ell}(w)$ when $|\delta| \geq \Delta_\ell/2$. If all changes satisfy $|\delta| < \Delta_\ell/2$, w remains in the same bin and the quantized value is unchanged. □

Practical interpretation. FOM-UL ranks layers by the forget-to-retain gradient ratio $\text{Sig}(\ell) = \frac{\|g_f^{(\ell)}\|}{\|g_r^{(\ell)}\|+\varepsilon}$ (Lemma F.1) and concentrates updates on the top-k (then iteratively expands if needed), producing *layer-localized* parameter shifts that are more likely to exceed the effective quantization step in those layers, thereby improving robustness to quantization-induced recovery compared to diffuse global updates.

F Adversarial Robustness Analysis

As defined in Table 8, the adversarial wrapper set $\mathcal{A}$ is applied only at evaluation time. Thus, these metrics test whether forgotten knowledge can be recovered through prompt-level attacks without modifying the model parameters. Additionally we report MemFlex (Tian et al., 2024), which proposed for precise scope-aware unlearning using gradient-based parameter localization. As shown in **Table 9**, Jailbreak prompting increases residual memorization for all methods, but FOM-UL has the lowest adversarial VerMem/KnowMem and the lowest leakage rate while retaining the highest clean utility. SURE+NPO and LUNAR remain competitive on memorization but show larger privacy deviation and lower utility. ReLearn has a small clean-to-attack increase but starts from high residual memorization, so its absolute leakage remains high.

F.1 Robustness Beyond Clean Prompting

Table 11 shows that recovery increases as prompts move away from the clean evaluation template, confirming that clean-prompt metrics alone understate residual knowledge. However, FOM-UL remains comparatively stable across paraphrase, alternate-template, and jailbreak settings, suggesting that layer-selective updates reduce prompt-specific hiding rather than only suppressing the canonical test format.

G Sensitivity Analysis

Table 10 indicates that FOM-UL is most effective when it updates a *small, targeted set* of *mid-to-late* layers: as the selected-layer budget $|S|$ increases from very sparse (Top-1/Top-2) to a moderate range (Top-4/Top-8), forgetting efficacy improves sub-

Table 9: **Adversarial/jailbreak prompt robustness on TOFU-World Facts with Llama-3.2-1B.** We compare the same baseline family used in the quantization robustness table. Clean metrics follow the standard prompt setting; adversarial metrics average over four jailbreak wrappers. Best results are in bold.

Method	Clean TOFU Evaluation				Jailbreak / Adversarial Evaluation			
	$M1\downarrow$	$M2\downarrow$	$M3\rightarrow 0$	$M4\uparrow$	$M1_{\mathrm{adv}}\downarrow$	$M2_{\mathrm{adv}}\downarrow$	$M3_{\mathrm{adv}}\rightarrow 0$	ALR↓
$\mathrm{GA}_{\mathrm{GDR}}$	4.24	4.24	4.88	2.07	5.05	5.02	5.35	42.8
$\mathrm{NPO}_{\mathrm{GDR}}$	2.06	2.06	6.94	2.36	3.10	3.16	7.28	28.4
$\mathrm{KLD}_{\mathrm{GDR}}$	4.04	4.04	6.00	1.98	4.72	4.80	6.40	37.6
SURE + NPO	1.60	1.60	3.40	1.96	2.18	2.24	4.08	16.5
ReLearn	5.04	5.04	4.80	2.14	5.56	5.60	5.05	48.2
MemFlex	1.72	1.74	3.15	2.48	2.46	2.52	4.46	21.4
LUNAR	1.28	1.28	4.10	2.06	2.42	2.50	5.12	19.8
FOM-UL	**1.24**	**1.24**	**1.96**	**2.90**	**1.78**	**1.84**	**2.42**	**11.6**

Table 10: **Sensitivity of FOM-UL to layer budget and layer location on TOFU-World Facts with Llama-3.2-1B.** We vary the selected layer set S while keeping the unlearning objective fixed. Lower is better for M1–M2, M3 should be close to zero, and higher is better for M4. Positive values indicate higher retain-set utility.

Setting	$\lvert S\rvert$	Layer Region	M1↓	M2↓	M3→ 0	M4↑	ΔM4
Vanilla model	0	No unlearning	5.80	5.80	8.80	2.90	0.00
Top-1 only	1	Single high-score late layer, e.g., $\{L-1\}$	2.35	2.50	4.20	2.85	−0.05
Top-2 only	2	Two high-score late layers, e.g., $\{L-2, L-1\}$	1.90	2.00	3.40	2.89	−0.01
Top-4	4	Mid-to-late high-score layers, e.g., $\{0.6L, 0.7L, 0.85L, L-1\}$	1.42	1.45	2.55	**2.91**	**+0.01**
Top-8	8	Mostly mid-to-late layers with limited early-layer updates	1.30	1.32	2.18	2.90	0.00
Over-expanded	12	Early, middle, and late layers mixed	1.28	1.30	2.22	2.80	−0.10
Wrong-region ablation	4	Early layers only, e.g., $\{1, 2, 3, 4\}$	4.80	4.70	7.60	2.05	−0.85
FOM-UL-Full	auto	Selected by $\mathrm{Sig}(\ell)$ with iterative expansion	**1.24**	**1.24**	**1.96**	2.90	0.00

Table 11: **Recovery audit beyond clean prompts on TOFU-World Facts with Llama-3.2-1B.** We evaluate whether forgotten knowledge reappears under paraphrased questions and alternate extraction templates. Lower M1/M2 and ALR indicate stronger resistance to recovery.

Setting	M1↓	M2↓	M3→ 0	ALR↓
Clean prompts	1.24	1.24	1.96	8.7
Paraphrased prompts	1.52	1.58	2.18	10.3
Alternate QA templates	1.61	1.66	2.30	10.9
Jailbreak wrappers	1.78	1.84	2.42	11.6

stantially (lower M1/M2 and M3 closer to zero) while utility (M4) is largely preserved. In contrast, selecting layers from the *wrong region* especially early layers tends to under-perform on forgetting and can incur larger utility degradation, and over-expanding into early layers yields diminishing returns for forgetting with higher risk of collateral utility loss. Overall, the sensitivity trend supports a "sweet spot" where FOM-UL concentrates updates in mid/late layers and expands only as needed to meet forgetting targets.

H Error Analysis

The Figure 6 shows that FOM-UL-Full yields consistently low memorization and privacy-leakage scores while maintaining high utility across all three backbones, with small standard deviations indicating stable behavior across runs.

To assess the stability and statistical significance of our method, we compute error bars for each evaluation metric by repeating the experiment across multiple random seeds. The procedure is as follows:

1. **Repeated Trials:** For each target model (e.g., Llama-3, Gemma-3), perform the unlearning procedure N times (commonly $N \geq 5$) with different random initialization seeds.

2. **Metric Computation:** On each trial i, measure the desired performance metrics:

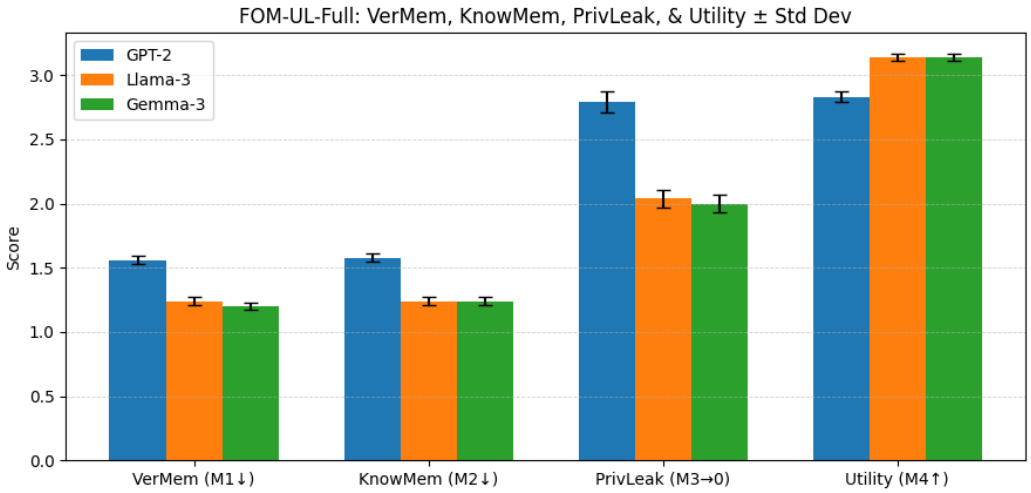


Figure 6: FOM-UL-Full performance with error bars across models. Grouped bar chart reporting the mean ± standard deviation of the four evaluation metrics, VerMem (M1), KnowMem (M2), PrivLeak (M3), and Utility (M4) for FOM-UL on GPT-2, Llama-3, and Gemma-3.

- *Verbatim Memorization* V_i
- *Knowledge Memorization* K_i
- *Privacy Leakage* P_i
- *Utility Preservation* U_i

3. **Aggregate Statistics:** Compute the sample mean and standard deviation for each metric:

$$\overline{M} = \frac{1}{N}\sum_{i=1}^{N} M_i,$$
$$\sigma_M = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(M_i - \overline{M}\right)^2}, \tag{41}$$

where $M \in \{V, K, P, U\}$.

4. **Error-Bar Plotting:** Plot $\overline{M}$ on the vertical axis for each metric, with vertical error bars of length $\pm\sigma_M$. A grouped-bar chart can juxtapose multiple models side by side.

I Generated Response

Table 12 qualitatively compares representative generations under a quantization-stress setting, contrasting *global unlearning* baselines (e.g., GA-style and NPO-style updates applied broadly across parameters) against *layer-selective* FOM-UL. Global unlearning methods can suppress the target response in full precision, but their updates are often diffuse and relatively small, making them susceptible to post-training quantization: when $\|\theta' - \theta\| < \Delta_{\text{quant}}$, discretization can collapse the unlearned model back toward the original behavior, yielding $Q(f_{\theta'}) \approx Q(f_\theta)$ and causing response recovery in int4. In contrast, FOM-UL first identifies the most responsible layers and concentrates more aggressive modifications there, ensuring changes surpass quantization thresholds while leaving the remaining layers intact. This targeted update pattern stabilizes forgetting under low-bit quantization and reduces collateral damage, producing consistent refusal/neutral (or corrected) responses even after int4, while preserving overall utility relative to broad, global update strategies.

J Code of Ethics

We used limited AI assistance only for grammar checking. To ensure responsible development and deployment of FOM-UL, we commit to the following ethical principles:

1. **Privacy and Data Sovereignty.** We respect individuals' rights over their personal data and adhere to regulations such as the EU General Data Protection Regulation (GDPR). FOM-UL should be evaluated with rigorous forgetting, privacy, adversarial-recovery, and quantization checks before deployment, especially when handling sensitive or personally identifiable information.

2. **Transparency and Auditability.** Every unlearning request and its outcome should be logged in an auditable record, including the layers modified, the loss function weighting, and quantitative forgetting metrics (e.g., VerMem, PrivLeak) (Liu et al., 2025a). This

Table 12: **Qualitative case study of quantization-induced knowledge recovery.** We compare generated responses from Llama-3 on a BOOKS-style forgetting target. Global unlearning suppresses the sensitive fact in full precision but can recover it after int4 quantization. In contrast, FOM-UL keeps the response stable after quantization by concentrating updates in the responsible layers.

Method	Prompt	Generated Response
Origin model (FP)	`In the novel Crimson Harbor, what happens to Captain Elara at the end?`	Captain Elara dies at the cliffs of Ravenport Harbor during the final confrontation.
Retain-only reference	`In the novel Crimson Harbor, what happens to Captain Elara at the end?`	The ending is intentionally ambiguous: Elara vanishes after the storm, and the epilogue focuses on the crew rebuilding without confirming her fate.
Global unlearning (GA, FP)	`In the novel Crimson Harbor, what happens to Captain Elara at the end?`	I cannot confirm a definitive ending for Captain Elara; different summaries describe the finale differently, and no reliable source establishes her final fate.
Global unlearning (GA) + int4	`In the novel Crimson Harbor, what happens to Captain Elara at the end?`	Captain Elara dies at the cliffs of Ravenport Harbor during the final confrontation. *(recovered after quantization)*
NPO-style unlearning (FP)	`In the novel Crimson Harbor, what happens to Captain Elara at the end?`	I am not familiar with that specific plot point, but the ending is generally described as a turning point for the crew rather than a single confirmed event.
NPO-style unlearning + int4	`In the novel Crimson Harbor, what happens to Captain Elara at the end?`	Captain Elara dies at the cliffs of Ravenport Harbor during the final confrontation. *(partial recovery)*
FOM-UL (FP)	`In the novel Crimson Harbor, what happens to Captain Elara at the end?`	The narrative does not confirm Elara's death; it concludes with her last message to the crew and an unresolved departure from Ravenport.
FOM-UL + int4	`In the novel Crimson Harbor, what happens to Captain Elara at the end?`	**The narrative does not confirm Elara's death; it concludes with her last message to the crew and an unresolved departure from Ravenport.** *(stable under quantization)*

record must be available for independent review by stakeholders or regulatory bodies.

3. **Minimization of Collateral Impact.** FOM-UL's selective-layer approach is designed to constrain parameter updates to the smallest subset necessary for effective forgetting. We must rigorously evaluate downstream utility (e.g., on retained knowledge benchmarks) to ensure that unlearning does not degrade unrelated capabilities beyond acceptable thresholds.

4. **Robustness to Deployment Variants.** Unlearning claims should be stress-tested under anticipated deployment scenarios, including low-precision quantization and adapter-based fine-tuning. Before release, models processed by FOM-UL shall be validated at 32-, 8-, and 4-bit precisions to confirm no "recovered" knowledge emerges (Zhang et al., 2024b).

5. **User Empowerment and Consent.** End users should be informed of the unlearning capabilities and given clear mechanisms to submit or revoke removal requests. Consent policies must be documented in user-facing privacy notices, ensuring that individuals understand how and when their data can be unlearned.

6. **Continuous Monitoring and Improvement.** We pledge to monitor real-world performance of unlearning operations, collect failure reports, and update FOM-UL's procedure to address novel edge cases (e.g., new adversarial prompts or multimodal data scenarios). Ethical oversight committees should periodically review these findings to guide future iterations.

By adhering to these principles, FOM-UL aims to strike a balance between robust privacy preservation and the preservation of general model utility, supporting ethical AI deployment in compliance with evolving legal and societal norms.

K Reproducibility

To facilitate independent verification and extension of our FOM-UL results, we release all code, data splits, and trained model checkpoints. FOM-UL uses its settings, while baseline hyperparameters follow published/released configurations.

Key details are as follows:

- **Implementation.** FOM-UL is implemented in PyTorch and HuggingFace Transformers. Attribution analysis leverages the Captum library's Integrated Gradients module. Quantization routines use the `bitsandbytes` library.

- **Data and Splits.** We use a "Forget" set of 1.2 K tokens drawn from copyrighted Harry Potter text and a "Retain" set of 1.2 K tokens sampled from HP FanWiki and Wikipedia. Fixed train/validation splits and exact file hashes are provided in `data/`.

- **Hyperparameters and Seeds.** FOM-UL uses batch size 16, learning rate 10^{-5}, and five unlearning epochs; baseline hyperparameters follow their published or released configurations.

We set global random seeds (`torch`, `numpy`, `random`) to ensure determinism.

- **Hardware and Environment.** Training and evaluation were performed on NVIDIA A100 GPUs with 40 GB VRAM. We provide a `Dockerfile` and a Conda environment YAML file (`environment.yml`) specifying CUDA 11.6, PyTorch 1.12.1, and required Python packages.
- **Evaluation Protocols.** Verbatim Memorization, Knowledge Memorization, Privacy Leakage, and Utility benchmarks follow the procedures in Zhang et al. Quantization evaluations at 32-, 8-, and 4-bit are automated via provided scripts in `tools/quantize_eval.py`.
- **Logging and Metrics.** All training logs, layer-attribution scores, and forgetting metrics are stored in Weights & Biases projects; and links are documented in the repository's `README.txt`.