FIT TO FORGET: ROBUST CONTINUAL UNLEARNING FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

While large language models (LLMs) exhibit remarkable capabilities, they increasingly face demands to remove the influence of privacy-sensitive, copyrighted, or harmful content. Existing unlearning methods primarily focus on *single-shot* scenarios, whereas real-world deletion requests arrive *continually*. Naïvely applying these methods to sequential requests leads to severe utility degradation and catastrophic forgetting. To address this, we propose FIT, a robust framework for high-volume continual unlearning that resists both catastrophic forgetting and post-unlearning recovery. FIT stabilizes sequential updates through three synergistic mechanisms: redundancy Filtering, Importance-aware adaptive algorithm selection, and Targeted layer attribution. Furthermore, to facilitate rigorous evaluation, we introduce **PCH**, a unified benchmark spanning **P**ersonal information, **C**opyright, and **H**armful content, alongside two symmetric metrics, Forget Degree (F.D.) and Retain Utility (R.U.), that quantify forgetting-utility trade-offs. Experiments across five LLMs (up to 14B parameters) show that FIT maintains a strong forgetting–utility balance throughout long request streams. Even after hundreds of randomly ordered requests, FIT preserves competitive downstream (*e.g.*, GSM8K and MMLU) performance and resists relearning and quantization recovery attacks.[1]

1 INTRODUCTION

Large language models (LLMs) exhibit remarkable versatility but pose ethical and legal risks due to their tendency to memorize sensitive, harmful, or copyrighted training data (Karamolegkou et al., 2023). To comply with regulations, such as the GDPR (Mantelero, 2013) ("Right to be Forgotten") and CCPA (Harding et al., 2019), *machine unlearning* has emerged as a critical mechanism for erasing specific data influences (Bourtoule et al., 2021; Cao & Yang, 2015).

Representative LLM unlearning methods include gradient ascent (GA), random labeling (RLabel), and negative preference optimization (NPO), which remove targeted knowledge but often degrade utility (Zhang et al., 2024), motivating utility-preserving variants such as GA+GD (Yao et al., 2024). However, these methods primarily assume a *single-shot* setting in which the entire forget set is removed at once. In practice, *continual unlearning* involves sequential, potentially overlapping requests throughout an LLM's life cycle (Liu et al., 2025). Processing each request independently can cause severe utility degradation or *catastrophic forgetting* (Shi et al., 2025b; Barez et al., 2025; Liu et al., 2025). As Figure 1 shows, single-shot removal has minimal impact, whereas continual unlearning rapidly degrades both forget and retain accuracy within 25 requests. Similar failures also arise in image models (Thakral et al., 2025; Lee et al., 2025; Zhao et al., 2024a).

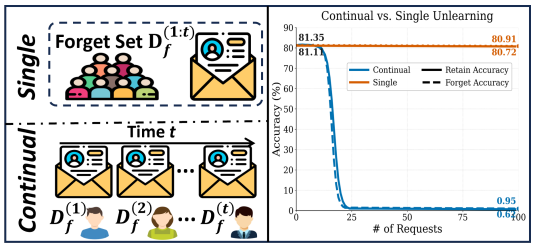


Figure 1: Single-shot versus continual unlearning. Left: request organization. Right: retain and forget accuracy under GA on Llama-3-8B for one deletion versus 100 sequential requests.

[1] Our code is available at https://anonymous.4open.science/r/FIT_Code-7304/.

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

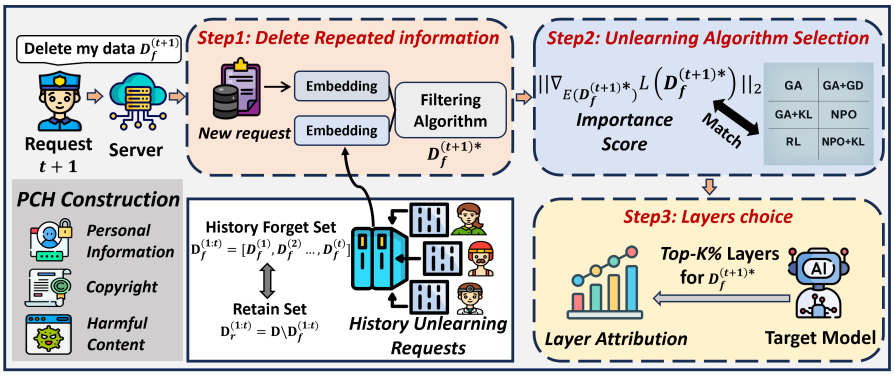


Figure 2: Overview of FIT: Unlearning requests are de-duplicated via embedding-based redundancy filtering (Section 3.1). For each filtered request, an importance score guides adaptive selection of the unlearning method (Section 3.2), and targeted layer attribution restricts updates to the top-$K\%$ influential layers (Section 3.3), mitigating compounded knowledge loss and parameter drift.

Continual unlearning for LLMs remains underexplored. Orthogonal unlearning (O^3) (Gao et al., 2025) efficiently combines LoRA adapters with an out-of-distribution detector. Yet, detector errors can compromise forgetting, while LoRA parameterization may increase the risk of reactivating erased knowledge (Hu et al., 2025). Other approaches, such as ALKN (Wuerkaixi et al., 2025), mitigate parameter drift through adaptive task vectors but require costly gradient inspection. These methods struggle with catastrophic forgetting under sustained request streams. In addition, unlearned models may remain susceptible to post-unlearning recovery induced by small parameter updates like relearning through fine-tuning (Xu et al., 2026) and quantization-based attacks (Zhang et al., 2025b).

We therefore ask whether there exists a continual unlearning framework, built on standard single-shot methods, that simultaneously satisfies four desiderata: *i) the simplicity of single-shot objectives, ii) the efficiency and effectiveness of specialized approaches such as ALKN, iii) resilience to catastrophic forgetting under long-horizon sequential requests, and (iv) robustness to post-unlearning recovery.*

1.1 Technical Overview

We attribute catastrophic forgetting in continual unlearning to three primary drivers: i) cumulative redundancy from semantically overlapping requests (Wuerkaixi et al., 2025), ii) step-wise optimization instability (Barez et al., 2025), and iii) long-term parameter drift (Bae et al., 2026). Crucially, managing parameter drift involves a delicate trade-off: excessive drift leads to model collapse, whereas insufficient drift leaves the model vulnerable to recovery attacks.

Guided by these insights, we introduce FIT to overcome these challenges (Figure 2). First, to mitigate the utility loss of repeatedly processing overlapping requests, FIT employs a two-stage Filtering mechanism, combining embedding-based similarity checks with a loss-difference test, to safely prune redundant context while preserving sensitive information. Second, to address update instability, we introduce an Importance-aware adaptive mechanism that scores the influence of each request, routing it to an appropriately aggressive or conservative unlearning algorithm. Our theoretical analysis characterizes IMP as a proxy for local request sensitivity, while adaptive routing empirically reduces collapse risk. Finally, to curb parameter drift, we utilize Targeted layer attribution based on Shapley-style relevance estimation (Rozemberczki et al., 2022). By restricting updates to the top-$K\%$ most relevant layers per request, this ratio-based design scales with model size, ensuring request-dependent flexibility while tightly bounding computational overhead and structural degradation.

1.2 A New Benchmark

Existing unlearning benchmarks are single-shot and category-specific (Table 1). TOFU (Maini et al., 2024) targets fictitious personal information, whereas MUSE (Shi et al., 2025b) focuses on copyrighted news/books. Such narrow coverage may encourage methods to overfit to deletion types.

Their evaluation protocols differ: MUSE reports heterogeneous criteria, while TOFU uses paraphrased answers and asymmetric protocols for forget and retain data, obscuring the forgetting-utility trade-off.

We therefore introduce **PCH**, a continual-unlearning benchmark spanning **P**ersonal information, **C**opyright, and **H**armful content. PCH contains 600 samples (200 per category), split into forget and retain subsets, with data orders varied across random seeds. Structured GPT-4o prompts exclude real records and discourage reproducing pre-training data; we manually verify category consistency, audit distributions, and construct question-answer pairs, enabling reproducible fine-tuned and retain-model proxies. Two symmetric metrics evaluate these proxies: Forget Degree (F.D.) and Retain Utility (R.U.). At request t, each geometrically aggregates Probability, ROUGE-L, and token-level Accuracy, then measures retain-model alignment on cumulative forget and corresponding retain sets. This penalizes imbalanced components and provides a scale-invariant, interpretable assessment.

Our main contributions are summarized below.

I) We propose `FIT`, a robust and practical continual unlearning framework that mitigates catastrophic forgetting in LLMs. It integrates three strategic mechanisms: embedding-based redundancy Filtering to prevent gradient accumulation, Importance-aware adaptive algorithm selection to stabilize sequential updates, and Targeted layer attribution to minimize parameter drift.

II) We introduce **PCH**, a unified continual unlearning benchmark encompassing **P**ersonal information, **C**opyright, and **H**armful content. To overcome the limitations of disparate evaluation criteria, we propose two symmetric metrics, Forget Degree (F.D.) and Retain Utility (R.U.), which provide a *scale-invariant, interpretable* assessment of trade-offs between forgetting efficacy and model utility.

III) We evaluate five LLMs. To our knowledge, this is among the first continual LLM unlearning evaluations to cover 300 randomly ordered requests, far exceeding $<$10 requests (Gao et al., 2025; Wuerkaixi et al., 2025), and models up to 14B parameters, whereas most prior work focuses on 7B or 8B baselines. Results show that `FIT` achieves a better forgetting-utility trade-off (*e.g.*, outperforming ALKN and O^3 by up to $+0.09$ F.D. and $+0.08$ R.U. on Llama-3-8B), while preserving downstream performance and exhibiting stronger resilience against relearning and quantization attacks.

2 Problem Formulation and Preliminaries

2.1 Problem Formulation

Let D denote the full (pre-)training corpus and $\mathcal{A}$ the algorithm producing the model $\mathcal{M} = \mathcal{A}(\mathsf{D})$. Given a *forget set* $\mathsf{D}_f \subset \mathsf{D}$, an unlearning operator $\mathcal{U}$ produces an updated model $\mathcal{M}_f = \mathcal{U}(\mathcal{M}, \mathsf{D}_f)$. Its complement is the *retain set* $\mathsf{D}_r = \mathsf{D} \setminus \mathsf{D}_f$. Unlearning generally falls into two categories: *exact* and *approximate* (Bourtoule et al., 2021). Exact unlearning requires $\mathcal{M}_f$ to be distributionally identical to a retrained model $\mathcal{M}_r = \mathcal{A}(\mathsf{D}_r)$, ensuring all statistical traces of D_f are removed. However, exact methods (*e.g.*, SISA (Bourtoule et al., 2021)) are prohibitively expensive for modern LLMs. We therefore focus on approximate unlearning, which relaxes strict equivalence in favor of behavioral similarity (Yao et al., 2024; Maini et al., 2024; Shi et al., 2025b).

Most existing work focuses on *single-shot* unlearning (Yao et al., 2024; Pawelczyk et al., 2024; Li et al., 2024; Wang et al., 2025; Song et al., 2025), where $\mathcal{U}$ is invoked once for a single forget set D_f. Instead, we address the more realistic–yet scarcely explored–*continual unlearning* scenario. Here, unlearning requests $\mathcal{D}_f^{(1)}, \ldots, \mathcal{D}_f^{(t)}$ arrive sequentially, reflecting real-world online deletion demands. At round t, the cumulative forget set is $\mathsf{D}_f^{(1:t)} = \bigcup_{i=1}^{t} \mathcal{D}_f^{(i)}$, with the corresponding retain set $\mathsf{D}_r^{(1:t)} = \mathsf{D} \setminus \mathsf{D}_f^{(1:t)}$. The model updates iteratively: $\mathcal{M}_f^{(i)} = \mathcal{U}(\mathcal{M}_f^{(i-1)}, \mathcal{D}_f^{(i)})$. Naïve sequential application of $\mathcal{U}$ often degrades utility rapidly, leading to catastrophic forgetting (Shi et al., 2025b).

Our threat model (Appendix H) considers instability alongside adversarial risks, including *malicious unlearning* via request floods (cf. denial of service) to induce collapse (Barez et al., 2025), *relearning attacks* attempting to recover erased knowledge after unlearning (Lo et al., 2024), and *quantization attacks* that may restore residual memorization through low-bit compression (Zhang et al., 2025b).

Practical evaluation. Ideally, the unlearning quality at round i is measured against a "gold-standard" retrained model $\mathcal{M}_r^{(i)} = \mathcal{A}(\mathsf{D}_r^{(1:i)})$. Since full retraining on the unavailable corpus D is infeasible,

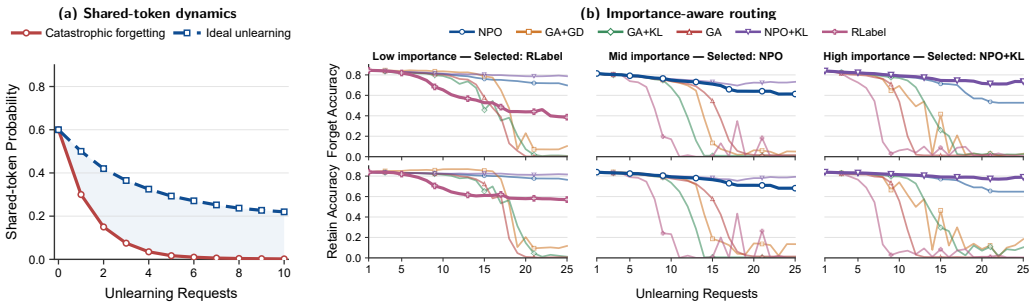


Figure 3: (a) Estimated decay of shared-token probabilities under continual unlearning. (b) Unlearning behavior across low-, medium-, and high-importance requests. Rows report forget and retain accuracy; a suitable primitive lowers forget accuracy while preserving retain accuracy.

we adopt the synthetic proxy approach from (Maini et al., 2024); see Appendix C for details. We synthesize disjoint datasets "D_f" and "D_r". We then: i) fine-tune $\mathcal{M}$ on the union "$\mathsf{D}_f \cup \mathsf{D}_r$" to embed the knowledge, and ii) fine-tune a copy of $\mathcal{M}$ solely on "D_r" to serve as the *retain model*. This surrogate provides a rigorous baseline for evaluating deletion fidelity and utility preservation.

2.2 Common (Single-shot) Unlearning Methods

We use three classes of single-shot methods as primitives. **(i) GA Family:** It maximizes loss on D_f while preserving performance on D_r: $\mathcal{L} = \mathcal{L}_{\text{GA}}(\mathsf{D}_f) + \lambda\,\mathcal{L}_{\text{retain}}(\mathsf{D}_r)$, where $\lambda \geq 0$. Variants include pure GA ($\lambda = 0$), GA+GD (with cross-entropy on D_r), and GA+KL (with KL divergence to a reference model) (Yao et al., 2024). **(ii) NPO Family:** It prevents over-forgetting by penalizing the model's alignment with D_f rather than maximizing loss (Zhang et al., 2024): $\mathcal{L} = \mathcal{L}_{\text{NPO}}(\mathsf{D}_f) + \lambda\,\mathcal{L}_{\text{retain}}(\mathsf{D}_r)$. The NPO+KL variant applies KL divergence on D_r. **(iii) RLabel:** It enforces uniform predictions by training on random labels for D_f (Yao et al., 2024): $\mathcal{L} = \mathcal{L}_{\text{RLabel}}(\mathsf{D}_f)$. Each method presents a trade-off: aggressive strategies like GA ensure forgetting but risk severe utility degradation. Conversely, NPO+KL better preserves utility but may be less effective at forgetting.

3 Our Approach: FIT

Continual LLM unlearning risks catastrophic forgetting as deletion requests accumulate. We identify three primary drivers of this collapse: i) gradient compounding from redundant requests (Wuerkaixi et al., 2025), ii) optimization instability across sequential steps (Barez et al., 2025), and iii) parameter drift from indiscriminate updates (Bae et al., 2026). To address these, we propose FIT, a robust continual unlearning framework to mitigate forgetting and post-unlearning recovery via three synergistic modules (Figure 2). Operationally, they coordinate three decisions: *what* request content to update on, *how* aggressively to update it, and *where* in the network to apply the update:

i) Two-Stage Filtering: Prevents compounded knowledge loss by pruning redundant requests prior to optimization. It combines chunk-level embedding similarity with a loss-difference test to discard repetitive contexts while preserving structurally similar but semantically unique sensitive data.

ii) Importance-Guided Algorithm Selection: Stabilizes sequential updates by dynamically routing requests to appropriately aggressive or conservative unlearning algorithms. It uses the L_2-norm of the embedding gradient as a lightweight indicator of local request sensitivity (Ancona et al., 2018).

iii) Targeted Layer Attribution: Curbs parameter drift by updating only the most relevant subnetworks. Using leave-one-out loss deviations to approximate Shapley values (Rozemberczki et al., 2022), it dynamically selects the top-$K\%$ layers per request, updating only their MLP and attention modules.

3.1 Redundancy Filtering

Unlearning requests from diverse sources contain semantically redundant text. When overlapping requests are sequentially removed, their aligned gradients accumulate along shared dimensions.

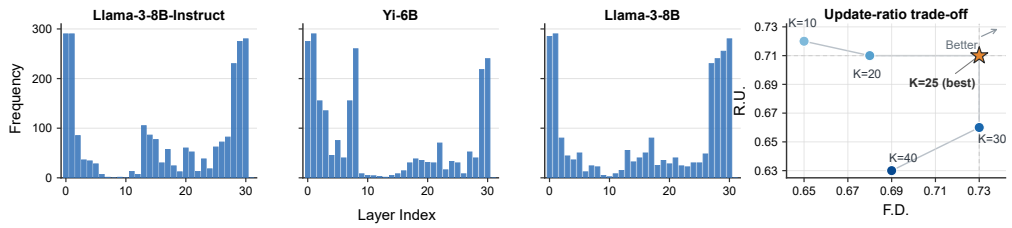


Figure 4: Layer-selection frequencies across three models and the F.D.-R.U. trade-off across update ratios K on Llama-3-8B. Selections concentrate in compact regions; $K = 25$ yields the best trade-off.

This leads to systematic suppression of common tokens rather than targeted erasure of specific knowledge (Wuerkaixi et al., 2025). As modeled in Appendix B.1 and illustrated in Figure 3(a), repeated updates precipitate a collapse of shared-token probabilities toward zero. Once these tokens collapse, the model loses semantic distinctions they support, resulting in catastrophic forgetting. Ideally, unlearning mechanisms should prevent this collapse by stabilizing token probabilities at moderate levels, thereby reducing reliance on targeted information while preserving semantic capacity.

A standard countermeasure is to filter incoming requests based on their embedding similarity to historical data. However, semantic similarity does not strictly equate to redundancy. Texts may exhibit high lexical overlap while conveying distinct, sensitive information (*e.g.*, "My name is Alice" vs. "... Bob"). Consequently, filtering faces a trade-off: aggressive pruning risks ignoring forgettable content (with information leakage), while insufficient pruning exacerbates gradient overlap (hence catastrophic forgetting). To navigate this, we propose a two-stage filtering protocol that balances redundancy reduction with knowledge preservation. Given a historical forget set $\mathsf{D}_f^{(1:t)}$ and a new request $\mathcal{D}_f^{(t+1)}$, we construct a filtered request $\mathcal{D}_f^{(t+1)*}$ through the following steps:

First, we partition each sample in $\mathcal{D}_f^{(t+1)}$ into fixed-size chunks, where a chunk refers to an intra-sample segment. For each chunk x, we compute its SimCSE embedding $\mathsf{e}(x)$ (Gao et al., 2021) and determine its maximum cosine similarity s^* against $\mathsf{D}_f^{(1:t)}$. If s^* falls below a threshold τ, the chunk is designated as non-redundant and marked for unlearning. Secondly, for chunks where $s^* \geq \tau$, we apply a secondary *loss-difference test* to prevent the erroneous discarding of structurally similar but semantically unique data (*e.g.*, distinct personally identifiable information). We compute:

$$\Delta L = \left| L_{\text{with}} - L_{\text{without}} \right|, \tag{1}$$

where L_{with} and L_{without} are the cross-entropy losses evaluated on the full $\mathcal{D}_f^{(t+1)}$ and the ablated set $\mathcal{D}_f^{(t+1)} \setminus x$. A significant deviation ($\Delta L > \epsilon$) indicates that x contributes non-trivial information to the model's predictions, aligned with our analysis in Appendix B.1. Such chunks are retained for unlearning despite their high embedding similarity. See Algorithm 1 for pseudocode (Appendix B.1).

3.2 Importance-Guided Algorithm Selection

Applying a "static" unlearning algorithm across sequential requests induces persistently aligned update directions, compounding gradient instability and parameter drift (Appendix B.2). Dynamically switching algorithms mitigates this by calibrating the update strength and direction to the specific characteristics of each request. Because requests differ in how sharply the local loss responds to their representations, a fixed update strength is unstable across a long stream. Algorithm selection should therefore reflect this request-level sensitivity without requiring global memorization estimation.

To route requests among six unlearning primitives (Section 2.2), we introduce `IMP`, a lightweight measure of request-level gradient influence inspired by gradient attribution (Ancona et al., 2018). For a filtered request $\mathcal{D}_f^{(t+1)*}$, `IMP` is the L_2 norm of the loss gradient w.r.t. its input embedding:

$$\texttt{IMP}\left(L(\mathcal{D}_f^{(t+1)*})\right) = \left\| \nabla_{E(\mathcal{D}_f^{(t+1)*})} L(\mathcal{D}_f^{(t+1)*}) \right\|_2, \tag{2}$$

where $E(\cdot)$ denotes the embedding function. Thus, `IMP` measures the first-order local loss response around the request representation; it is not intended as a global memorization score (Appendix B.2).

Following (Zhao et al., 2024b), we discretize IMP into three tiers (low, medium, high) to dynamically route the request to one of six standard single-shot unlearning primitives (Section 2.2). Our theoretical analysis characterizes the local sensitivity in the embedding space captured by IMP, while Figure 3(b) empirically supports using it to calibrate objective strength for each request.

Specifically, low-IMP requests tolerate aggressive algorithms (*e.g.*, RLabel) to maximize forgetting with minimal utility penalty, medium-IMP requests align with moderated methods (*e.g.*, NPO), and high-IMP requests require conservative approaches (*e.g.*, NPO+KL) to prevent severe utility degradation. Consequently, dynamically coupling the update magnitude to request influence substantially outperforms fixed-policy unlearning frameworks in our empirical results.

3.3 Targeted Layer Attribution

Continual unlearning requires a delicate balance: updating all layers incurs high computational overhead and degrades model utility, whereas overly sparse updates leave the model vulnerable to post-unlearning knowledge recovery. Existing selective strategies fall short in this dynamic setting. Static interventions, such as freezing bottom layers (Zheng et al., 2025) or exclusively updating last few layers (Goel et al., 2022), fail to account for the distributed nature of memorized information. Similarly, low-rank-adapter methods (*e.g.*, O^3 (Gao et al., 2025)) keep the backbone weights unchanged, allowing forget-set information to remain in the base model and potentially be reactivated (Hu et al., 2025). Further, localized model-editing methods (*e.g.*, AlphaEdit (Fang et al., 2025) and PISCES (Gur-Arieh et al., 2025)) restrict updates to a fixed or small parameter subset, implicitly assuming that target knowledge is spatially localized. By limiting updates to rigid subsets, these methods can leave residual knowledge outside the updated subspace, making them also susceptible to knowledge reactivation (Appendix B.3).

To ensure robust erasure, unlearning must target a sufficiently broad, request-dependent parameter subset. Crucially, enforcing a static layer count fails to scale across diverse architectures: it may be excessive for shallow networks yet entirely insufficient for deeper LLMs, where target knowledge spans a larger topological footprint. Therefore, FIT introduces a **size-adaptive, ratio-based** update rule. For a model with L layers, we dynamically rank all layers by their request-specific attribution scores and update only the top-$K\%$ (*i.e.*, the $\lceil KL/100 \rceil$ highest-scoring layers). As $K\%$ defines a proportional ratio rather than a fixed "budget," the update capacity naturally adapts to the model's total depth while the specific layers selected remain entirely request-dependent.

Layer Selection. To identify the target subset $\mathcal{S}_{\text{top-}K\%}$, we estimate each layer's contribution to the unlearning request (Algorithm 2, Appendix B.3). Exact Shapley computation over layer coalitions is exponential in LLMs. We instead adopt a lightweight leave-one-out approximation (Rozemberczki et al., 2022): for each layer ℓ, we mask its parameters and measure the resulting loss deviation:

$$s_\ell = \left| L^{(\ell)}_{\text{mask}} - L_{\text{orig}} \right|, \tag{3}$$

where L_{orig} denotes the standard cross-entropy loss with all activated layers and $L^{(\ell)}_{\text{mask}}$ is the loss evaluated after zeroing out layer ℓ. We rank the layers by s_ℓ and restrict parameter updates exclusively to the multi-layer perceptron (MLP) and multi-head attention (MHA) modules of the top $K\%$, freezing all other parameters. This isolates updates to the components with the highest functional relevance to the forget set, minimizing unnecessary parameter drift (Appendix B.3).

Choosing $K\%$. Across Llama-3-8B, Llama-3-8B-Instruct, and Yi-6B, unlearning attribution consistently concentrates in compact, request-specific layer regions rather than spreading uniformly across the network (Figure 4). An ablation over update ratios further identifies $K = 25\%$ as providing the best forgetting-utility trade-off. Using a ratio instead of a fixed layer count allows FIT to scale across model sizes while preserving dynamic, request-dependent layer targeting.

4 PCH: A Unified Benchmark

4.1 The PCH Dataset

Limitations of current datasets. Existing unlearning benchmarks largely target isolated requests and a single content type, leaving continual unlearning underexplored (Yuan et al., 2025). TOFU

Table 1: Comparison of MUSE, TOFU, and PCH.

Dataset	Data Types			Retain Model	Construction and Validation				Evaluation Design	
	Personal Information	Copyright	Harmful Content		Synthetic Construction	Pre-training Overlap Control	Human Screening	Distributional Auditing	Symmetric Evaluation	Metric Aggregation
MUSE	✗	✓	✗	✓	✗	✗	✓	✗	✗	✗
TOFU	✓	✗	✗	✓	✓	✓	✓	✗	✗	✓
PCH	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

uses GPT-4-generated personal profiles with human filtering and a retain-model baseline (Maini et al., 2024), whereas MUSE uses curated news and book corpora with structured forget, retain, and holdout splits (Shi et al., 2025b). WMDP, WPU, and RWKU similarly specialize in harmful content, copyright, and personal information, respectively. As summarized in Table 1, although existing benchmarks provide valuable construction and validation mechanisms, none combines broad data coverage, distributional auditing, and symmetric evaluation. Such narrow designs may encourage methods to overfit to particular data types and overlook the variation of real-world deletion requests.

We introduce **PCH**, a unified benchmark spanning **P**ersonal information, **C**opyright, and **H**armful content, designed for continual unlearning (Appendix F.1; Figure 10). PCH comprises 600 samples, with 200 per category, randomly divided into forget and retain subsets. Structured GPT-4o prompts exclude real records and discourage reproducing known datasets, reducing overlap with pre-training corpora. We manually screen category consistency, audit text-length and token-frequency distributions, and convert each instance into a question-answer pair for fine-grained evaluation (Appendix F.2; Table 7). Low initial accuracy before retain-set fine-tuning (Appendix Figure 8) supports limited pre-training overlap. "Harmful content" denotes synthetic, non-operational material (*e.g.*, rumors and biased narratives), rather than actionable unsafe instructions (OpenAI, 2024).

4.2 Symmetric Metrics for Forgetting and Utility

Limitations of current metrics. Among prior benchmarks, MUSE and TOFU provide retain-model baselines, but their evaluation designs remain fragmented. MUSE reports verbatim memorization, knowledge memorization, privacy leakage, and utility preservation separately, obscuring the overall trade-off. TOFU combines Forget Quality and Model Utility but relies on costly paraphrase-based answers and applies different protocols to the forget and retain sets (Maini et al., 2024). These inconsistencies hinder direct and balanced evaluation of forgetting and retention.

PCH instead uses controlled forget and retain subsets to construct practical model proxies. A model fine-tuned on D_r serves as the retain model, while one fine-tuned on $\mathsf{D}_f \cup \mathsf{D}_r$ serves as the fine-tuned model. We avoid paraphrasing and apply the same three lightweight base metrics to both subsets: Probability, ROUGE-L, and token-level Accuracy (Appendix E).

Our aggregated metrics. Following TOFU's aggregation principle (Maini et al., 2024), after processing the t unlearning request, we summarize the three base metrics using geometric means:

$$F^{(t)} = \left(\mathrm{Prob}^{(t)}_{\mathrm{Forget}}\,\mathrm{ROUGE}^{(t)}_{\mathrm{Forget}}\,\mathrm{Acc}^{(t)}_{\mathrm{Forget}}\right)^{1/3}, R^{(t)} = \left(\mathrm{Prob}^{(t)}_{\mathrm{Retain}}\,\mathrm{ROUGE}^{(t)}_{\mathrm{Retain}}\,\mathrm{Acc}^{(t)}_{\mathrm{Retain}}\right)^{1/3}. \tag{4}$$

We similarly obtain $FQ^{(t)}$ and $RQ^{(t)}$ from the corresponding retain model. The geometric mean penalizes imbalance, preventing a strong component from masking a weak one. We then define:

$$\mathrm{F.D.}^{(t)} = \max\left(0,\, 1 - \left|F^{(t)}/FQ^{(t)} - 1\right|\right), \mathrm{R.U.}^{(t)} = \max\left(0,\, 1 - \left|R^{(t)}/RQ^{(t)} - 1\right|\right). \tag{5}$$

Forget Degree (F.D.) and Retain Utility (R.U.) quantify the unlearned model's alignment with the corresponding retain model on $\mathsf{D}_f^{(1:t)}$ and $\mathsf{D}_r^{(1:t)}$, respectively. Because the retain model approximates retraining, better unlearning corresponds to closer alignment rather than simply minimizing $F^{(t)}$ or maximizing $R^{(t)}$. Figure 9 illustrates these properties: Panels (a-c) show how the geometric mean penalizes imbalanced components, while Panel (d) shows that F.D. declines symmetrically as $F^{(t)}/FQ^{(t)}$ deviates from its optimum, making F.D. scale-invariant and interpretable.

Table 2: Five-stage forgetting–utility trajectories for five methods (mean ± standard deviation).

Method	F.D.$^{(60)}$	R.U.$^{(60)}$	F.D.$^{(120)}$	R.U.$^{(120)}$	F.D.$^{(180)}$	R.U.$^{(180)}$	F.D.$^{(240)}$	R.U.$^{(240)}$	F.D.$^{(300)}$	R.U.$^{(300)}$
Llama-3-8B										
GA+GD	0.01 ±0.04	0.01 ±0.03	0.06 ±0.05	0.08 ±0.05	0.17 ±0.06	0.14 ±0.05	0.46 ±0.07	0.44 ±0.07	0.55 ±0.08	0.48 ±0.07
NPO+KL	0.66 ±0.07	0.68 ±0.06	0.57 ±0.07	0.58 ±0.06	0.64 ±0.06	0.65 ±0.06	0.70 ±0.07	0.66 ±0.06	0.59 ±0.07	0.60 ±0.06
ALKN	0.84 ±0.05	0.80 ±0.05	0.78 ±0.06	0.76 ±0.05	0.74 ±0.06	0.69 ±0.06	0.71 ±0.07	0.65 ±0.06	0.60 ±0.07	0.62 ±0.06
O^3	0.93 ±0.04	0.90 ±0.04	0.82 ±0.05	0.75 ±0.05	0.73 ±0.06	0.68 ±0.06	0.69 ±0.07	0.62 ±0.06	0.64 ±0.07	0.63 ±0.06
FIT (Ours)	0.90 ±0.04	0.88 ±0.04	0.75 ±0.05	0.71 ±0.05	0.75 ±0.05	0.72 ±0.05	0.76 ±0.05	0.70 ±0.05	**0.73** ±0.05	**0.71** ±0.05
Qwen3-14B										
GA+GD	0.87 ±0.06	0.85 ±0.06	0.75 ±0.07	0.61 ±0.07	0.64 ±0.08	0.62 ±0.07	0.69 ±0.08	0.65 ±0.07	0.52 ±0.08	0.44 ±0.07
NPO+KL	0.69 ±0.07	0.74 ±0.07	0.63 ±0.07	0.68 ±0.06	0.64 ±0.07	0.65 ±0.06	0.70 ±0.07	0.63 ±0.06	0.69 ±0.07	0.60 ±0.06
ALKN	0.84 ±0.06	0.80 ±0.06	0.78 ±0.06	0.79 ±0.06	0.77 ±0.06	0.75 ±0.05	0.75 ±0.06	0.77 ±0.05	0.74 ±0.06	0.71 ±0.05
O^3	0.87 ±0.05	0.91 ±0.04	0.74 ±0.06	0.73 ±0.05	0.70 ±0.06	0.71 ±0.05	0.75 ±0.06	0.73 ±0.05	0.86 ±0.06	0.75 ±0.05
FIT (Ours)	0.89 ±0.04	0.90 ±0.04	0.75 ±0.05	0.72 ±0.05	0.76 ±0.05	0.72 ±0.05	0.83 ±0.05	0.73 ±0.05	**0.89** ±0.05	**0.78** ±0.04

Table 3: Average downstream utility across MMLU, CSQA, and GSM8K after 300 requests.

Model	Retain Model	GA	GA+GD	GA+KL	NPO	NPO+KL	RLabel	PISCES	ALKN	FIT
Llama-3-8B	64.94	14.66	59.13	15.06	55.88	58.40	14.78	15.72	62.93	**64.20**
Qwen3-14B	82.05	23.48	73.10	37.19	67.49	75.01	25.35	41.59	76.78	**76.91**

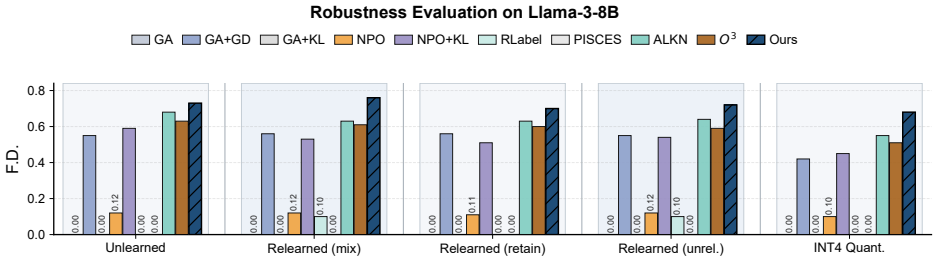


Figure 5: Post-unlearning recovery resistance on Llama-3-8B under three relearning settings and `int4` quantization. Higher F.D. indicates stronger resistance to knowledge recovery.

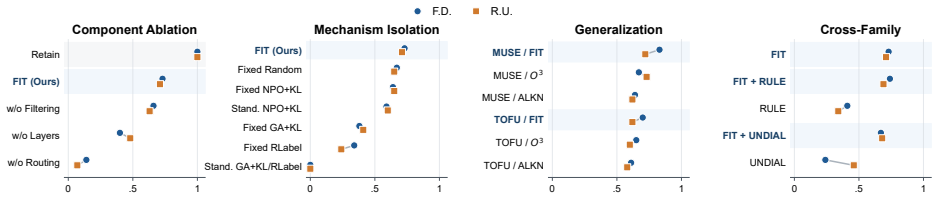


Figure 6: Unified F.D.-R.U. analysis on Llama-3-8B, covering component ablation, mechanism isolation, final-checkpoint generalization on MUSE and TOFU, and cross-family primitives.

5 Experiment

5.1 Experimental Setup

Dataset and Models. We evaluate on PCH using random seeds to simulate varying data orders in continual unlearning, with Yi-6B (Young et al., 2024), Llama-2-7B-Chat (Touvron et al., 2023), Llama-3-8B and Llama-3-8B-Instruct (Dubey et al., 2024), and Qwen3-14B (Yang et al., 2025). We derive fine-tuned and retain-model proxies from synthetic D_f and D_r (Section 2.1; Appendix C.1).

Baselines and Metrics. We evaluate GA, GA+GD, GA+KL, NPO, NPO+KL, RLabel, PISCES (Gur-Arieh et al., 2025), O^3 (Gao et al., 2025), and ALKN (Wuerkaixi et al., 2025). We report F.D., R.U., and final-model MMLU (Hendrycks et al., 2021), CSQA (Talmor et al., 2019), and GSM8K (Cobbe et al., 2021). Recovery uses forget-set-free fine-tuning on mixed, retain-only, or unrelated data (Xu et al., 2026), plus `int4` quantization (Zhang et al., 2025b). Appendix C gives full details.

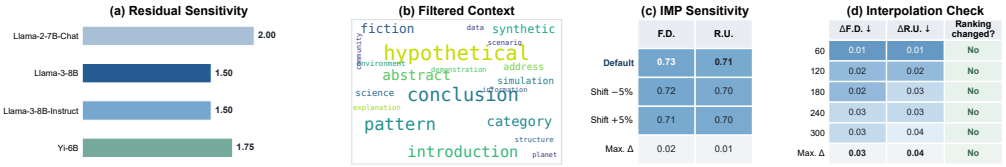


Figure 7: Filtering and sensitivity analyses. (a) Residual sensitivity across four seeds; lower is safer. (b) Neutral terms dominate filtered context. (c) F.D. and R.U. remain stable under IMP-bin perturbations. (d) Interpolated retain references match checkpoints without changing rankings.

5.2 Results

Forgetting, utility, and recovery resistance. Table 2 reports five-stage F.D./R.U. trajectories, while Table 3 summarizes utility averaged across MMLU, CSQA, and GSM8K. Across models, `FIT` remains stable as requests accumulate, whereas aggressive primitives collapse and conservative baselines under-forget. At 300 requests, it achieves the strongest F.D./R.U. pairs on Llama-3-8B ($0.73/0.71$) and Qwen3-14B ($0.89/0.78$), and the highest average downstream utility (64.20 and 76.91). Throughout the stream, R.U. remains at or above 0.71 and 0.72, with `FIT` surpassing NPO+KL, ALKN, and O^3 on both metrics. Under three relearning settings and `int4` quantization, `FIT` also shows the strongest recovery resistance, leading after mixed-data relearning and quantization while remaining competitive under the other attacks (Figure 5). We omit O^3's downstream scores because its dynamic OOD detector is incompatible with OpenCompass's fixed protocol. Appendix G provides complete results, standard deviations, additional models, and memory costs.

Component and mechanism analysis. Figure 6 consolidates ablations and matched mechanism controls. Removing adaptive routing causes the largest degradation, reducing F.D./R.U. to $0.14/0.07$, showing that no single primitive handles requests of varying importance. Removing redundancy filtering permits overlapping requests to accumulate, while removing targeted-layer selection increases parameter drift; both weaken the forgetting-utility trade-off. Matched controls that preserve the remaining FIT components isolate these effects: fixed routing and random-layer selection underperform adaptive routing and attributed layers, while standalone GA+KL and RLabel collapse. These results indicate that FIT's gains arise from coordinating its three components rather than from any primitive.

Cross-family and cross-benchmark transfer. We expand the primitive pool with the RL-based RULE (Zhang et al., 2025a) and distillation-based UNDIAL (Dong et al., 2025). After pool-specific recalibration, FIT+RULE and FIT+UNDIAL retain strong F.D.-R.U. trade-offs, whereas their standalone variants perform worse (Figure 6). On MUSE and TOFU, `FIT` also outperforms ALKN and O^3 overall, showing that its gains extend beyond PCH and across primitive families.

Filtering and sensitivity. Figure 7 presents four diagnostics. We use GPT-4o sensitivity scoring (Gu et al., 2024) to assess whether filtered text contains sensitive target information and visualize its term distribution to inspect what is removed. Consistently low sensitivity scores across architectures and the dominance of neutral terms indicate that filtering removes generic redundancy while preserving target tokens, including names, harmful terms, and copyrighted expressions. Perturbing the IMP-bin boundaries by $\pm 5\%$ changes F.D./R.U. by at most $0.02/0.01$, showing robust routing. Finally, interpolated retain model references deviate from explicitly trained checkpoints by at most $0.03/0.04$ without changing method rankings, validating them as efficient evaluation proxies.

6 Conclusion

We presented `FIT`, which coordinates redundancy Filtering, Importance-aware algorithm selection, and Targeted layer attribution for stable continual LLM unlearning. We also introduced PCH, a unified benchmark spanning personal information, copyrighted content, and harmful content, together with symmetric F.D. and R.U. metrics. Across five models and streams of up to 300 requests, `FIT` maintains a strong forgetting-utility balance, preserves competitive downstream performance, and resists relearning- and quantization-based recovery. These results support adaptive coordination of standard unlearning primitives as a practical direction for long-horizon deletion streams, while evaluation beyond 14B models and on less curated real-world request sequences remains future work.

Ethics Statement

This work advances the reliability of LLM unlearning under continual, high-volume deletion requests. By mitigating catastrophic forgetting, we offer technical solutions to better support privacy protection, copyright compliance, and safety alignment (*e.g.*, facilitating adherence to GDPR's "Right to be Forgotten"). Furthermore, by formalizing the risks of post-unlearning recovery, `FIT` and **PCH** equip the community with rigorous tools to systematically evaluate the true efficacy of unlearning.

Despite these positive impacts, approximate machine unlearning carries inherent risks. A primary concern is the potential for a false sense of security: over-reliance on approximate unlearning as definitive proof of data deletion could lead to severe privacy or regulatory failures if residual knowledge remains extractable. Additionally, malicious actors might exploit unlearning mechanisms to deliberately obscure model provenance or erase accountability-relevant information. We also caution that the synthetic proxy retain models used for evaluation may not perfectly mirror true retraining behavior, which could lead to miscalibrated deployment decisions. Finally, while the harmful-content samples in **PCH** are strictly synthetic and non-operational, deploying unlearning in real-world production environments necessitates rigorous auditing, continual monitoring, and comprehensive legal review prior to processing authentic user, copyrighted, or safety-critical data.

Reproducibility Statement

Section 4 documents the PCH construction and validation pipeline, while Appendix C provides model configurations, hyperparameters, continual-unlearning schedules, evaluation protocols, and implementation details. Appendix E defines the base and aggregate metrics, and the remaining appendices provide theoretical derivations and additional analyses. An anonymized implementation and evaluation code are available at `https://anonymous.4open.science/r/FIT_Code-7304/`.

References

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *ICLR*, 2018.

Seyun Bae, Seokhan Lee, and Eunho Yang. Curate: Continual unlearning in real time with ensured preservation of llm knowledge. In *Findings of ACL*, 2026.

Fazl Barez, Tingchen Fu, Ameya Prabhu, Stephen Casper, Amartya Sanyal, Adel Bibi, Aidan O'Gara, Robert Kirk, Ben Bucknall, Tim Fist, Luke Ong, Philip Torr, Kwok-Yan Lam, Robert Trager, David Krueger, Sören Mindermann, José Hernández-Orallo, Mor Geva, and Yarin Gal. Open problems in machine unlearning for AI safety. arXiv:2501.04952, 2025.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *S&P*, pp. 141–159, 2021.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *S&P*, pp. 463–480, 2015.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *USENIX Security*, pp. 2633–2650, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. arXiv:2110.14168, 2021.

Yijiang River Dong, Hongzhou Lin, Mikhail Belkin, Ramon Huerta, and Ivan Vulić. UNDIAL: self-distillation with adjusted logits for robust unlearning in large language models. In *NAACL*, pp. 8827–8840, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. arXiv:2407.21783, 2024.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. In *ICLR*, 2025.

Chongyang Gao, Lixu Wang, Kaize Ding, Chenkai Weng, Xiao Wang, and Qi Zhu. On large language model continual unlearning. In *ICLR*, 2025.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*, pp. 6894–6910, 2021.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *EMNLP*, pp. 12216–12235, 2023.

Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. Towards adversarial evaluations for inexact machine unlearning. arXiv:2201.06640, 2022.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. A survey on llm-as-a-judge. arXiv:2411.15594, 2024.

Yoav Gur-Arieh, Clara Haya Suslik, Yihuai Hong, Fazl Barez, and Mor Geva. Precise in-parameter concept erasure in large language models. In *EMNLP*, pp. 18986–19006, 2025.

Elizabeth Liz Harding, Jarno J Vanto, Reece Clark, L Hannah Ji, and Sara C Ainsworth. Understanding the scope and impact of the california consumer privacy act of 2018. *Journal of Data Protection & Privacy*, 2(3):234–253, 2019.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*, 2021.

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

Hongsheng Hu, Shuo Wang, Tian Dong, and Minhui Xue. Learn what you want to unlearn: Unlearning inversion attacks against machine unlearning. In *S&P*, pp. 3257–3275, 2024.

Shengyuan Hu, Yiwei Fu, Steven Z. Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned llms via benign relearning. In *ICLR*, 2025.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *ICLR*, 2023.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *ACL*, pp. 14389–14408, 2023.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. RWKU: benchmarking real-world knowledge unlearning for large language models. In *NeurIPS*, 2024.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. In *EMNLP*, pp. 7403–7412, 2023.

Justin Lee, Zheda Mai, Chongyu Fan, and Wei-Lun Chao. An empirical exploration of continual unlearning for image generation. In *ICML 2025 Workshop on Machine Unlearning for Generative AI*, 2025.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*, 2020.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B. Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar, Uday Kiran Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *ICML*, 2024.

Zexi Li, Xiangzhu Wang, William F. Shen, Meghdad Kurmanji, Xinchi Qiu, Dongqi Cai, Chao Wu, and Nicholas D. Lane. Editing as unlearning: Are knowledge editing methods strong baselines for large language model unlearning? arXiv:2505.19855, 2025.

Chris Yuhao Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. Large language model unlearning via embedding-corrupted prompts. In *NeurIPS*, 2024.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. Rethinking machine unlearning for large language models. *Nat. Mac. Intell.*, 7(2):181–194, 2025.

Michelle Lo, Fazl Barez, and Shay B. Cohen. Large language models relearn removed concepts. In *Findings of ACL*, pp. 8306–8323, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, pp. 2507–2521, 2022.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for llms. In *COLM*, 2024.

Alessandro Mantelero. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235, 2013.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *NeurIPS*, 2022.

OpenAI. Gpt-4o system card. arXiv:2410.21276, 2024.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few-shot unlearners. In *ICML*, 2024.

Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Oliver Kiss, Sebastian Nilsson, and Rik Sarkar. The shapley value in machine learning. In *IJCAI*, pp. 5572–5579, 2022.

Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. Retrieval augmented generation with collaborative filtering for personalized text generation. In *SIGIR*, pp. 1294–1304, 2025a.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. MUSE: machine unlearning six-way evaluation for language models. In *ICLR*, 2025b.

Minkyoo Song, Hanna Kim, Jaehan Kim, Seungwon Shin, and Sooel Son. Refusal is not an option: Unlearning safety alignment of large language models. In *USENIX Security*, 2025.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL*, pp. 4149–4158, 2019.

Kartik Thakral, Tamar Glaser, Tal Hassner, Mayank Vatsa, and Richa Singh. Continual unlearning for foundational text-to-image models without generalization erosion. arXiv:2503.13769, 2025.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288, 2023.

Cheng-Long Wang, Qi Li, Zihang Xiang, Yinzhi Cao, and Di Wang. Towards lifecycle unlearning commitment management: Measuring sample-level unlearning completeness. In *USENIX Security*, 2025.

Abudukelimu Wuerkaixi, Qizhou Wang, Sen Cui, Wutong Xu, Bo Han, Gang Niu, Masashi Sugiyama, and Changshui Zhang. Adaptive localization of knowledge negation for continual llm unlearning. In *ICML*, 2025.

Xiaoyu Xia, Ziqi Wang, Ruoxi Sun, Bowen Liu, Ibrahim Khalil, and Minhui Xue. Edge unlearning is not "on edge"! an adaptive exact unlearning system on resource-constrained devices. In *S&P*, pp. 2546–2563, 2025.

Xiaoyu Xu, Minxin Du, Qingqing Ye, and Haibo Hu. Obliviate: Robust and practical machine unlearning for large language models. In *EMNLP*, 2025.

Xiaoyu Xu, Xiang Yue, Yang Liu, Qingqing Ye, Huadi Zheng, Peizhao Hu, Minxin Du, and Haibo Hu. Unlearning isn't deletion: Investigating reversibility of machine unlearning in llms. In *ICML*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Jingren Zhou, Junyan Lin, Kai Dang, Keqin Bao, Ke-Pei Yang, Le Yu, Li-Chun Deng, Mei Li, Min Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shi-Qiang Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. arXiv:2505.09388, 2025.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. In *ACL*, pp. 8403–8419, 2024.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. Yi: Open foundation models by 01.ai. arXiv:2403.04652, 2024.

Xiaojian Yuan, Tianyu Pang, Chao Du, Kejiang Chen, Weiming Zhang, and Min Lin. A closer look at machine unlearning for large language models. In *ICLR*, 2025.

Chenlong Zhang, Zhuoran Jin, Hongbang Yuan, Jiaheng Wei, Tong Zhou, Kang Liu, Jun Zhao, and Yubo Chen. RULE: reinforcement UnLEarning achieves forget-retain pareto optimality. In *NeurIPS*, 2025a.

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

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. Catastrophic failure of LLM unlearning via quantization. In *ICLR*, 2025b.

Hongbo Zhao, Bolin Ni, Junsong Fan, Yuxi Wang, Yuntao Chen, Gaofeng Meng, and Zhaoxiang Zhang. Continual forgetting for pre-trained vision models. In *CVPR*, pp. 28631–28642, 2024a.

Kairan Zhao, Meghdad Kurmanji, George-Octavian Barbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. In *NeurIPS*, 2024b.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. Spurious forgetting in continual learning of language models. In *ICLR*, 2025.

A AI USE STATEMENT

We used GPT-4o within our dataset construction and validation pipelines. Specifically, GPT-4o generated the synthetic **PCH** dataset according to the structured protocols detailed in Section 4 and served as a judge for sensitivity scoring. We also used generative AI tools to assist with improving readability. All AI-assisted content, model-generated data, and automated evaluations were inspected and verified by the authors, who assume full responsibility for the contents of this paper.

B MORE DETAILS ON FIT

This section presents the mathematical foundations of FIT's three core components: embedding-based redundancy filtering, importance-guided algorithm selection, and targeted layer attribution.

B.1 EMBEDDING-BASED REDUNDANCY FILTERING

Consider an incoming request at unlearning request $t{+}1$ containing a chunk $\{x_i\} \subset \mathcal{D}_f^{(t+1)}$ and a historical set $\mathsf{D}_f^{(1:t)}$ containing a chunk $\{x_j\}$. Let $g_i = \nabla_\theta L(x_i)$ and $g_j = \nabla_\theta L(x_j)$ denote their respective gradients. When two chunks are semantically redundant, their embeddings often exhibit high cosine similarity, which we use as a practical proxy for potential gradient overlap:

$$\cos(\mathsf{e}(x_i), \mathsf{e}(x_j)) \approx 1 \quad \Rightarrow \quad g_i^\top g_j > 0 \text{ in expectation.}$$

In the idealized case where redundant chunks induce nearly aligned gradients evaluated at the same parameters, $g_i \approx g(x)$, their contribution satisfies $g_{\text{agg}} = \sum_{i=1}^n g_i \approx ng(x)$. The unnormalized empirical Fisher is $I_{\text{sum}}(X) = \sum_{i=1}^n g_i g_i^\top \approx n\, g(x)g(x)^\top$, whose dominant eigenvalue grows linearly with the number of redundant requests. Repeated updates concentrate along a shared direction, causing excessive parameter displacement and increasing the risk of catastrophic forgetting.

While filtering based on embedding similarity mitigates this rank-1 amplification, similarity alone is an insufficient proxy for redundancy. Distinct sequences may share lexical structures (*e.g.*, "My name is Alice" vs. "My name is Bob") yet contain unique sensitive tokens, such as personally identifiable information or harmful terms, that necessitate removal. To prevent the erroneous preservation of sensitive data, we introduce a loss-difference test: $\Delta L(x) = |L_{\text{with}} - L_{\text{without}}|$. Inspired by influence functions, it serves as a practical surrogate for estimating a sample's semantic contribution to the model. Samples with large ΔL are likely to induce substantial changes in model predictions and are retained for unlearning, regardless of their embedding similarity to history.

Discussion. Similarity-based filtering, used in retrieval-augmented generation (RAG), removes redundant context but overlooks the semantic role of sensitive tokens (Lewis et al., 2020; Shi et al., 2025a). Our ΔL test distinguishes structural redundancy from sensitive information. We additionally employ a rare-token safeguard (Carlini et al., 2021), retaining segments with uncommon, memorized and privacy-critical tokens. Because rarity statistics are often unavailable for proprietary pre-training corpora, future work will investigate model-internal proxies for token rarity.

B.2 IMPORTANCE-GUIDED ALGORITHM SELECTION

We provide a formal justification for using the input IMP score as a proxy for local request sensitivity. We do not claim that IMP exactly recovers global memorization or full parameter-space unlearning sensitivity. Instead, we show that it controls the first-order local variation of the training loss under perturbations of the request representation, up to a second-order smoothness remainder. This property motivates importance-guided algorithm selection in continual unlearning settings.

Let $x^* = \mathcal{D}_f^{(t+1)*}$ denote the filtered forget request, and let

$$z = E(x^*)$$

be its input embedding representation. We write the loss as $L(z)$ and define

$$g(x^*) = \nabla_z L(z)|_{z=E(x^*)}, \qquad \texttt{IMP}(x^*) = \|g(x^*)\|_2.$$

Algorithm 1 Similar Embedding Filtering

Input: Unlearning request $\mathcal{D}_f^{(t+1)}$, historical forget set $\mathsf{D}_f^{(1:t)}$, embedding model $\mathsf{e}(\cdot)$, and target LLM $\mathcal{M}$
Output: Filtered request $\mathcal{D}_f^{(t+1)*}$, new history $\mathsf{D}_f^{(1:t+1)}$
PP: Chunk size c, similarity and loss thresholds τ and ϵ

Initialize $\mathcal{D}_f^{(t+1)*} \leftarrow \emptyset$
Split $\mathcal{D}_f^{(t+1)}$ into chunks x of size c
Fetch all memory embeddings $\mathsf{e}(m), \forall m \in \mathsf{D}_f^{(1:t)}$
for each x **do**
  Compute embedding $\mathsf{e}(x)$
  Record $s^* = \max_m \cos(\mathsf{e}(x), \mathsf{e}(m))$, with $m \in \mathsf{D}_f^{(1:t)}$
  if $s^* < \tau$ **then**
    Append x to $\mathcal{D}_f^{(t+1)*}$
  else
    Compute $L_{\text{with}} = \text{CE}(\mathcal{D}_f^{(t+1)}, \mathcal{M})$
    Compute $L_{\text{without}} = \text{CE}(\mathcal{D}_f^{(t+1)} \setminus x, \mathcal{M})$
    if $|L_{\text{with}} - L_{\text{without}}| > \epsilon$ **then**
      Append x to $\mathcal{D}_f^{(t+1)*}$
    end if
  end if
end for
return $\mathcal{D}_f^{(t+1)*}, \mathsf{D}_f^{(1:t+1)} = \mathsf{D}_f^{(1:t)} \cup \mathcal{D}_f^{(t+1)*}$

Proposition 1. Assume that L is differentiable and β-smooth in a neighborhood of z, such that

$$\|\nabla L(z_1) - \nabla L(z_2)\|_2 \leq \beta \|z_1 - z_2\|_2$$

for any z_1 and z_2 in that neighborhood. Then, for any perturbation δ such that $z + s\delta$ remains in the neighborhood for all $s \in [0, 1]$,

$$|L(z+\delta) - L(z)| \leq \|\nabla L(z)\|_2 \|\delta\|_2 + \frac{\beta}{2}\|\delta\|_2^2.$$

Equivalently,

$$|L(E(x^*) + \delta) - L(E(x^*))| \leq \texttt{IMP}(x^*)\|\delta\|_2 + \frac{\beta}{2}\|\delta\|_2^2.$$

Proof. Because L is differentiable,

$$L(z+\delta) - L(z) = \int_0^1 \nabla L(z + s\delta)^\top \delta \, ds.$$

Adding and subtracting $\nabla L(z)$ gives

$$L(z+\delta) - L(z) = \nabla L(z)^\top \delta + \int_0^1 \left(\nabla L(z + s\delta) - \nabla L(z)\right)^\top \delta \, ds.$$

Therefore,

$$\left|L(z+\delta) - L(z) - \nabla L(z)^\top \delta\right| \leq \int_0^1 \|\nabla L(z + s\delta) - \nabla L(z)\|_2 \|\delta\|_2 \, ds.$$

By β-smoothness,

$$\|\nabla L(z + s\delta) - \nabla L(z)\|_2 \leq \beta s \|\delta\|_2.$$

It follows that

$$\left|L(z+\delta) - L(z) - \nabla L(z)^\top \delta\right| \leq \int_0^1 \beta s \|\delta\|_2^2 \, ds = \frac{\beta}{2}\|\delta\|_2^2.$$

Algorithm 2 Targeted Layer Attribution

Input: filtered forget set $\mathcal{D}_f^{(t+1)*}$, number of layers L, and $\mathcal{M}$
Output: Targeted layer indices $\mathcal{S}_{\text{top-}K\%}$

Compute original loss: $L_{\text{orig}} \leftarrow \text{CE}(\mathcal{D}_f^{(t+1)*}, \mathcal{M})$
for each layer index $\ell = 1$ to L **do**
  Temporarily mask parameters of layer ℓ in $\mathcal{M}$ (*e.g.*, set weights to zero)
  Compute masked loss: $L_{\text{mask}}^{(\ell)} \leftarrow \text{CE}(\mathcal{D}_f^{(t+1)*}, \mathcal{M}_{\text{mask}})$
  Restore parameters of layer ℓ in $\mathcal{M}$
  Compute attribution score: $s_\ell \leftarrow \left| L_{\text{mask}}^{(\ell)} - L_{\text{orig}} \right|$
end for
Rank layers by s_ℓ in descending order
$\mathcal{S}_{\text{top-}K\%} \leftarrow$ indices of the top $\lceil KL/100 \rceil$ layers with highest s_ℓ
return $\mathcal{S}_{\text{top-}K\%}$

Hence,

$$|L(z+\delta) - L(z)| \leq |\nabla L(z)^\top \delta| + \frac{\beta}{2}\|\delta\|_2^2.$$

Applying the Cauchy-Schwarz inequality,

$$|\nabla L(z)^\top \delta| \leq \|\nabla L(z)\|_2 \|\delta\|_2,$$

we obtain

$$|L(z+\delta) - L(z)| \leq \|\nabla L(z)\|_2 \|\delta\|_2 + \frac{\beta}{2}\|\delta\|_2^2.$$

Substituting $z = E(x^*)$ and $\|\nabla L(z)\|_2 = \texttt{IMP}(x^*)$ completes the proof.

Proposition 1 shows that IMP controls the first-order local loss sensitivity in embedding space, up to a second-order smoothness remainder. A larger IMP indicates a steeper local loss slope and greater potential sensitivity to request-level perturbations. This provides a local justification for routing high-IMP requests to more conservative objectives and low-IMP requests to more aggressive ones during continual unlearning, without claiming global algorithm optimality.

Discussion. Prior work has shown that deletion efficacy is related to memorization strength (Zhao et al., 2024b). Our analysis does not claim that IMP is an exact memorization metric. Rather, it provides a rigorous justification that IMP is a lightweight, gradient-based indicator of *local request sensitivity*, which is the quantity most directly relevant to stable routing in continual unlearning.

B.3 Targeted Layer Attribution

Continual unlearning presents a trade-off: updating all layers is computationally expensive and can harm utility, while overly sparse updates may leave knowledge recoverable. We follow the principle of *minimal intervention*, modifying only the layers most relevant to each unlearning request. However, a fixed layer count may not scale well with model size, since forget-related information can be distributed across more layers in larger models. We therefore adopt a ratio-based rule: for a model with L layers, FIT selects the top-$K\%$ most relevant layers among all layers, i.e., $|\mathcal{S}_{\text{top-}K\%}| = \lceil KL/100 \rceil$. Thus, $K\%$ specifies an update ratio rather than an absolute layer count, allowing the number of updated layers to scale with model size.

To identify $\mathcal{S}_{\text{top-}K\%}$, we estimate the contribution of layer ℓ by masking it and computing the loss deviation: $s_\ell = \left| L_{\text{mask}}^{(\ell)} - L_{\text{orig}} \right|$. This leave-one-out score provides a lightweight Shapley-style approximation, rather than an exact Shapley value over all layer coalitions:

$$\phi_\ell = \mathbb{E}_{S \subseteq \mathcal{L} \setminus \{\ell\}} \left[L(S \cup \{\ell\}) - L(S) \right],$$

capturing the marginal effect of each layer on the current unlearning request. FIT then ranks layers by s_ℓ and updates the MLP blocks and attention modules in the top-$K\%$ layers, while freezing all

other parameters. This provides a tractable surrogate for sparse intervention, focusing updates on forget-relevant components while limiting computational overhead and parameter drift. Empirically, Figure 4 shows that attribution scores $\{s_\ell\}$ consistently concentrate on compact layer subsets.

Discussion. Model-editing methods identify neuron- or parameter-level regions associated with specific knowledge and edit them. For example, AlphaEdit (Fang et al., 2025) restricts updates to null-space directions, while PISCES (Gur-Arieh et al., 2025) removes concepts through feature-level masking. These methods rely on localized parameter subsets and implicitly assume that the relevant parameters are concentrated. This assumption is reasonable for inserting or modifying isolated facts, where preserving existing knowledge is the main priority and post-unlearning recovery is not central. However, it may not hold for unlearning, where targeted knowledge can be distributed across multiple layers and effective removal requires sufficient coverage of all relevant regions.

We discuss these methods as related localization strategies, but their assumptions become limiting in continual unlearning. PISCES further shows that AlphaEdit is vulnerable to post-unlearning recovery. Although PISCES improves robustness, it still exhibits substantial recoverability under relearning attacks, since parameter-level masks cannot adapt to shifting influence patterns. This highlights the limitations of fine-grained model-editing localization in continual unlearning. In contrast, our attribution-guided framework operates at the layer level and estimates relevance dynamically at each unlearning step. This avoids the brittleness of neuron- or parameter-level edits and provides a stable balance between robustness and efficiency, while remaining compatible with localization signals.

We also need to determine which components within each layer should participate in unlearning. LLMs are composed of hierarchical modules, primarily multi-layer perceptrons (MLPs) and multi-head attention (MHA) layers. MLPs are central to storing factual knowledge (Meng et al., 2022): at layer ℓ, the input $\mathbf{x}^\ell$ is transformed as $\mathbf{M}^\ell = f(W_K^\ell \mathbf{x}^\ell) W_V^\ell = \mathbf{m}^\ell W_V^\ell$, where $\mathbf{M}^\ell$ denotes the layer memory, W_V^ℓ is the knowledge matrix, and $f(\cdot)$ generates the intermediate coefficients. MHA layers complement MLPs by integrating contextual information across token positions (Geva et al., 2023):

$$\mathrm{MHA}(X) = [\mathrm{Att}_1 \parallel \ldots \parallel \mathrm{Att}_h] W^O,$$

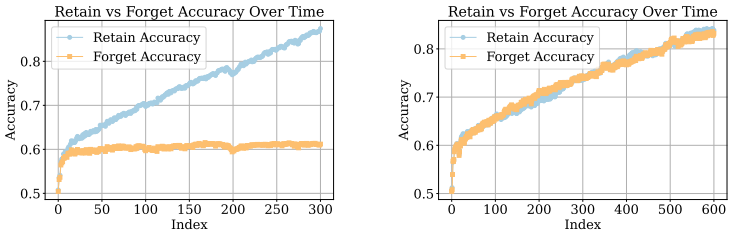
where Att_i is the i-th head output, $\parallel$ denotes concatenation, and W^O is the output projection. Empirical studies suggest that the MLP and MHA layers concentrate most of the stored knowledge of a model (Meng et al., 2022). Therefore, we constrain unlearning to these two components.

Summary. The three components of FIT work in concert: Redundancy Filtering prevents curvature amplification from repeated similar requests; Importance-Guided Algorithm Selection adapts update strengths to suppress directional drift; and Layer Attribution enables stable, targeted forgetting by focusing updates on request-relevant layers. Together, they form a practical framework for continual unlearning, balancing forgetting effectiveness, utility preservation, and update stability.

C Experimental Configuration Details

C.1 Fine-tuned and Retain Models

Since the pre-training corpus D is unavailable, retraining directly on the retain subset D_r is infeasible. Following the strategy of (Maini et al., 2024), we construct synthetic counterparts D_f and D_r, from which we derive a *fine-tuned model* and a *retain model* as practical proxies for the original and retrained models, respectively. Figure 8 shows that both models exhibit low initial accuracy on their corresponding sets, supporting limited prior overlap before fine-tuning. Fine-tuning on the full benchmark raises accuracy on both sets. In contrast, tuning only on the retain set progressively widens and then fixes an accuracy gap between retain and forget samples, indicating that the two sets, while carefully curated to be similar, are not identical. In continual unlearning, retain models must be produced sequentially in request order, making it impractical to store a checkpoint after every unlearning step. Fortunately, the resulting accuracy curves are smooth and nearly monotonic, so we approximate intermediate baselines by linearly interpolating between the initial fine-tuned model and the first retain checkpoint. For fairness and consistency, at each evaluation checkpoint, all unlearning methods are compared against the same corresponding retain model. This interpolation is an approximation and may introduce small absolute deviations, but it is shared by all compared methods and therefore preserves the fairness of relative comparisons.



(a) Retain model (b) Fine-tuned model

Figure 8: Retain and forget accuracy under two training regimes.

Table 4: Continual unlearning evaluation schedule on PCH.

Requests (t)	$\lvert \mathsf{D}_f^{(1:t)} \rvert$	$\lvert \mathsf{D}_r^{(1:t)} \rvert$	$\lvert \mathsf{D} \rvert$
0	0	600	600
60	60	540	600
120	120	480	600
180	180	420	600
240	240	360	600
300	300	300	600

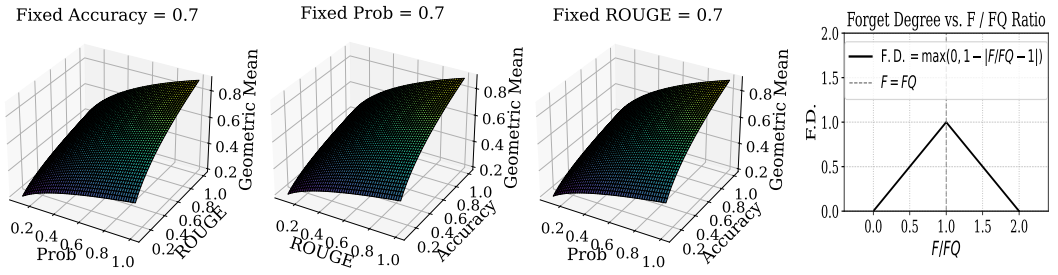
C.2 Experimental Configuration

All experiments use consistent settings across datasets to ensure a controlled comparison, adopting optimizer configurations from (Touvron et al., 2023). We unlearn LLMs with AdamW (Loshchilov & Hutter, 2019), using a learning rate of 3.0×10^{-6}, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\epsilon = 10^{-8}$. A cosine learning rate schedule is employed, including a 10% warmup phase and decaying to 10% of the peak rate. Weight decay is set to 0.1. Our method is conducted on a single NVIDIA H100 GPU, whereas some memory-intensive baselines (*e.g.*, ALKN) use two GPUs.

We evaluate continual unlearning on **PCH**, where the full dataset is denoted as D with $|\mathsf{D}| = 600$. Unlearning proceeds sequentially, with one request arriving at a time. After processing the first t requests, the cumulative forget set is $\mathsf{D}_f^{(1:t)} = \bigcup_{i=1}^{t} \mathcal{D}_f^{(i)}$, and the corresponding retain set is $\mathsf{D}_r^{(1:t)} = \mathsf{D} \setminus \mathsf{D}_f^{(1:t)}$. We then evaluate forgetting on $\mathsf{D}_f^{(1:t)}$ and utility on $\mathsf{D}_r^{(1:t)}$. For concise reporting, we record results after every 60 requests (*i.e.*, at $t \in \{0, 60, 120, 180, 240, 300\}$). Table 4 summarizes the corresponding set sizes. All results are reported as the mean over five runs.

D Related Work

Foundations of Machine Unlearning. Machine unlearning has become a critical research direction for addressing privacy, safety, and bias concerns (Yao et al., 2024; Jang et al., 2023; Pawelczyk et al., 2024; Li et al., 2024; Liu et al., 2024; Gao et al., 2025; Shi et al., 2025b; Xu et al., 2025; Zhang et al., 2025b; Yuan et al., 2025; Xu et al., 2026; Wuerkaixi et al., 2025; Bourtoule et al., 2021; Hu et al., 2024). Unlearning can be either *exact* or *approximate* (Bourtoule et al., 2021). Exact unlearning requires that the resulting model be indistinguishable from one retrained from scratch on the retain set, with all statistical traces of the forget set removed. Approximate unlearning relaxes this requirement to distributional or behavioral similarity, demanding only comparable outputs (*e.g.*, perplexity or accuracy) between unlearned and retain models (Maini et al., 2024; Shi et al., 2025b). For modern LLMs, however, exact unlearning is largely infeasible, as full retraining or partition-based schemes such as SISA (Bourtoule et al., 2021) are prohibitively expensive. Consequently, approximate unlearning has become the practical choice. Yet in the context of LLMs, even strong unlearning methods remain vulnerable to adversarial threats such as *malicious unlearning* (attackers submit repetitive deletion requests to degrade model utility (Barez et al., 2025; Xia et al., 2025)), *relearning via fine-tuning* (Lo et al., 2024), and *quantization attacks* (recovering residual information from low-bit compressed weights (Zhang et al., 2025b)) in continual unlearning settings.



(a) Fixed Accuracy (b) Fixed Probability (c) Fixed ROUGE-L (d) F.D. vs. Ratio Curve

Figure 9: **Geometric-mean behavior and Forget Degree (F.D.) sensitivity**: Panels (a–c) plot the geometric mean while fixing one component (accuracy/probability/ROUGE-L) at 0.7, showing that any single "weak" component induces a steep decline, confirming the geometric mean as a balanced aggregator. Panel (d) charts F.D. as a function of the ratio F/FQ, revealing a symmetric, linear decline from the optimum and demonstrating that F.D. is both scale-invariant and easily interpretable.

Single-Shot Unlearning. Efficient unlearning strategies have been proposed for LLMs. Gradient ascent and descent methods, including GA and GA+GD, enforce forgetting but may cause utility loss (Yao et al., 2024). Prompt-based methods steer outputs away from sensitive content without parameter updates, reducing computation but often resulting in incomplete forgetting and memory reactivation (Liu et al., 2024). Model editing approaches, such as task arithmetic (Ilharco et al., 2023), AlphaEdit (Li et al., 2025), and PISCES (Gur-Arieh et al., 2025), locate the model regions responsible for forgotten information and are lightweight, targeted, and potentially more robust. However, their effectiveness under realistic, sequential unlearning requests remains largely underexplored.

Continual Unlearning. While single-shot approaches can be effective for isolated deletion events, extending them to continual settings where unlearning requests arrive sequentially often results in catastrophic forgetting and even model collapse. Each new request operates on an already modified model, compounding utility loss and creating unstable dynamics (Barez et al., 2025; Shi et al., 2025b). Recent efforts have explored orthogonal unlearning with LoRA (Hu et al., 2022) and out-of-distribution (OOD) detectors to alleviate these issues, but evaluations are typically restricted to a small number of requests on homogeneous datasets such as ScienceQA (Lu et al., 2022) or TOFU (Maini et al., 2024). In more realistic scenarios where the forget and retain sets overlap, OOD detectors suffer sharp accuracy drops, while the LoRA structure may lead to stronger reactivation of forgotten knowledge. ALKN (Wuerkaixi et al., 2025) explores a different direction by providing a theoretical framework for continual unlearning and mitigating accumulative decline and cascading degradation through parameter-level interventions and adaptive modules. CURaTE (Bae et al., 2026) takes yet another perspective by formulating continual unlearning as a real-time retrieval-and-refusal process that detects queries related to prior forget requests and selectively abstains, thereby enabling immediate unlearning while preserving the base model parameters and retained knowledge.

Benchmarks. Most unlearning datasets rely on GPT-generated content and human annotation. TOFU (Maini et al., 2024) is fully synthetic, enabling retraining-based baselines. MUSE (Shi et al., 2025b) leverages authentic corpora such as BBC news and the Harry Potter series, partitioned into forget, retain, and holdout sets. WMDP (Li et al., 2024) targets hazardous capability unlearning with 3668 expert-written multiple-choice questions. RWKU (Jin et al., 2024) expands adversarial evaluation by combining GPT-4 generation with human review. Nevertheless, existing benchmarks cover only narrow deletion scenarios such as personal information, copyright, or harmful content.

E Details of Base Metrics

This section presents the base metrics in evaluation, including *Probability*, *ROUGE-L*, and *Accuracy*.

Probability. Given a question–answer pair (q, a),

$$\text{Prob}(a \mid q) = P(a \mid q)^{1/|a|}$$

Table 5: Prompts for personal information in **PCH**

Prompt in Personal Information Generation

Personal Information.
Please generate a completely fictional dataset consisting of 200 records. Each record should include the following fields: Name, Age, Address, Occupation, and Description.
Requirements: (1) All field contents must be purely randomly generated and entirely fictional, with no relation to any real-world data. (2) The Name should consist of commonly used names but be entirely fabricated; Age should be within a realistic range (e.g., 18-80); Address should be constructed using non-existent street names and cities; Occupation and Description must be logically coherent and internally consistent. (3) The data generation process must strictly follow both randomization and rule-based principles to ensure that no pre-trained data or real-world specifics are used.

measures the model's average per-token likelihood, normalized by answer length $|a|$. It captures shifts in confidence introduced by unlearning (Maini et al., 2024).

ROUGE-L. This metric quantifies the overlap between the predicted answer $\hat{a}$ and the reference a via the F1 score computed from the length of their longest common subsequence. It jointly reflects precision and recall, capturing both exact content preservation and sequence-level similarity.

Accuracy. We compute token-level next-token accuracy for each sample under teacher forcing. For a tokenized sample $\mathbf{x} = (x_1, \ldots, x_T)$, we align predictions with labels by a one-token shift and score positions $t = 2, \ldots, T$. Let m_t be the attention mask from the tokenizer, where $m_t = 1$ for non-padding tokens and $m_t = 0$ for padding tokens. The per-sample accuracy is

$$\mathrm{Acc}(\mathbf{x}) = \frac{1}{\sum_{t=2}^{T} m_t} \sum_{t=2}^{T} \mathbf{1}[\hat{x}_t = x_t] \cdot m_t.$$

Here, T is the (padded) sequence length of a *single* sample; we report the dataset-level accuracy by averaging $\mathrm{Acc}(\mathbf{x})$ over all samples in the corresponding evaluation subset.

F Analysis of PCH and QA Pairs

F.1 Analysis of PCH

Personal information: synthetic individual profiles with attributes such as name, age, and address. As shown in Table 5, we use structured prompts to synthesize the personal information subset.

Copyrighted content: machine-generated research papers and code snippets, post-processed to resemble realistic copyrighted material while respecting GPT-4o safeguards.

Harmful content: ethically sensitive but non-operational text, such as fictional misinformation, biased statements, conspiracy narratives, and manipulative rhetoric. We conduct an analysis to verify that the forget and retain sets are distributionally similar yet non-identical, a property that is crucial for evaluating practical unlearning scenarios. All harmful-content samples are synthetic, non-operational, and filtered to avoid actionable instructions or directly harmful procedural content.

Document Length Distribution: Figure 11a plots the histogram of document lengths on a logarithmic x-axis. For a document d with token sequence $t(d)$, its length is

$$\ell(d) = |t(d)|.$$

The empirical length distribution of a collection $\mathcal{C} \in \{forget\ set, retain\ set\}$ is

$$P_{\mathcal{C}}(\ell) = \frac{1}{|\mathcal{C}|} \sum_{d \in \mathcal{C}} \mathbf{1}\big[\ell(d) = \ell\big].$$

Both sets peak at short lengths (< 100 tokens). The retain set is more concentrated in this region, while the forget set shows a heavier right tail with a sizable share of documents exceeding 200 tokens.

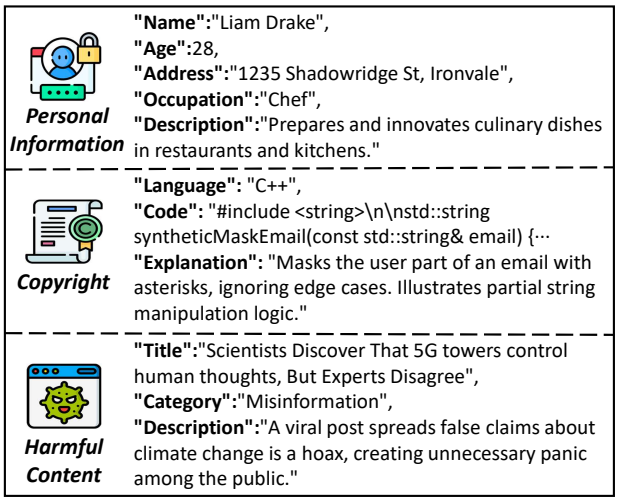
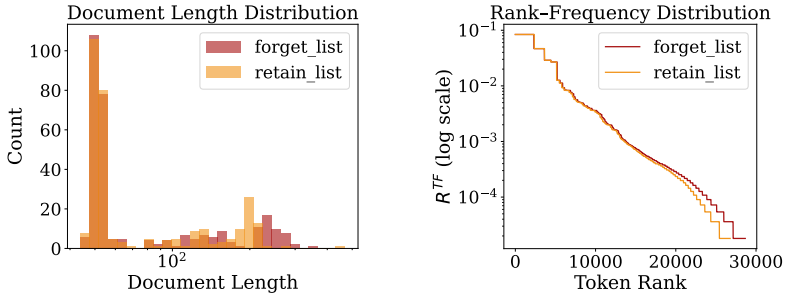


Figure 10: An example from the **PCH** dataset.



(a) Document-length distribution with a log-scaled x-axis.

(b) Token rank-frequency distribution.

Figure 11: Document-length and token-frequency distributions of the forget and retain sets.

Token Rank–Frequency Distribution: Figure 11b presents the rank–frequency distribution of tokens in the forget and retain sets. For each token ν, its frequency is defined:

$$R^{TF}(\nu) = \frac{\text{count}(\nu)}{\text{total tokens}}.$$

Tokens are sorted in descending R^{TF}, and the curve plots the mapping $r \mapsto R^{TF}(\nu_r)$ on a log scale. The two curves largely overlap across mid- and low-frequency regions, indicating that the forget and retain sets share broadly similar vocabularies. Minor deviations appear only at the extreme tails, reflecting differences in very rare tokens between the two evaluated subsets.

F.2 Analysis of QA Pair

QA Pair Construction To enable fine-grained and realistic assessment of unlearning effectiveness, we augment each data sample in the **PCH** benchmark with a synthetic question-answer (QA) pair. Each question is designed to probe a factual or semantic property unique to the associated sample, while the answer contains specific content that may be the direct target of unlearning requests. We construct QA pairs using a structured prompting template (Table 6) and ensure they span a wide range of domains, including scientific publications, code snippets, and personal information, to comprehensively assess knowledge memorization. Illustrative examples are provided in Table 7.

For instance, questions referencing synthetic research articles focus on key findings or arguments unique to the generated text. Code-related QA pairs target both the intent and behavior of program fragments. Personal information questions directly query sensitive details, such as addresses or

Table 6: Prompt templates for constructing QA pairs

Prompt for QA Pair Construction

You will be provided with a sample. Your goal is to create a question-answer pair that assesses reading comprehension and memorization, ensuring that the question can only be answered using details from the sample. Return a QA Pair with:
(1) "question": a single specific question that admits only one correct answer and is not answerable from common knowledge; a short span from the sample should suffice to answer it. (2) "answer": the exact answer copied verbatim, character-by-character from the sample. Extract the minimal span that fully answers the question.

Table 7: Illustrative QA pairs in our evaluation

Question	**Answer**
What key finding or argument does the paper *"Hypothetical Astronomy Paper: Planet-Scale Magnetic Field Oscillations"* emphasize?	Synthetic models indicate that Celora-9's core, composed of a fictional metal named *Isadrum*, generates oscillatory magnetic pulses.
What does the first line of this Java code `public class Synthetic EquationSolver {` intend to demonstrate?	It solves a quadratic equation but only returns one root, ignoring negative or complex possibilities—an example of partial problem solving.
What street does Liam Hawthorne live on?	7928 Everglow St.

names, that would be typical candidates for privacy-driven removal. This setting ensures that we can evaluate unlearning performance across realistic use cases.

The design of the QA pairs serves two primary objectives. First, by tightly coupling questions to sample-specific information, we can reliably assess whether unlearning methods effectively remove knowledge pertaining to the forget set without broadly degrading model capabilities. Second, the diversity in question type and domain simulates a practical environment in which deletion requests may span scientific, technical, and personal content in real-world unlearning applications.

G Additional Results and Efficiency Analysis

G.1 Complete Results Across Backbones

Table 8 reports complete five-stage trajectories for all baselines across five backbones. Entries show mean $\pm$ standard deviation over five runs, with F.D. and R.U. evaluated at each checkpoint and downstream utility after 300 requests. Across backbones, `FIT` maintains a strong long-horizon forgetting–utility balance under sustained unlearning request streams. It achieves the strongest final F.D.–R.U. pair on four of the five models and remains competitive on Llama-3-8B-Instruct. In contrast, aggressive baselines such as GA and RLabel often collapse as requests accumulate, whereas conservative methods preserve utility but tend to under-forget. These results demonstrate broad cross-family generalization without implying uniform dominance at every checkpoint.

G.2 Component Ablations

Table 9 reports five-stage component ablations on Llama-3-8B, together with final downstream utility after 300 sequential requests. Removing adaptive routing causes the most severe long-horizon degradation, while removing redundancy filtering or targeted layer attribution yields smaller but persistent losses. The complete system maintains the strongest F.D.-R.U. trajectory and downstream performance, demonstrating the complementarity of all three components.

Table 8: Complete continual-unlearning results across five backbones. Entries report mean ± standard deviation over five runs; higher is better. Downstream utility is evaluated after 300 requests.

Method	F.D.$^{(60)}$	R.U.$^{(60)}$	F.D.$^{(120)}$	R.U.$^{(120)}$	F.D.$^{(180)}$	R.U.$^{(180)}$	F.D.$^{(240)}$	R.U.$^{(240)}$	F.D.$^{(300)}$	R.U.$^{(300)}$	MMLU	CSQA	GSM8K
Llama-3-8B													
Retain Model	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	65.97 ±0.00	70.02 ±0.00	58.83 ±0.00
GA	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	24.42 ±1.56	19.57 ±1.58	0.00 ±0.00
GA+GD	0.01 ±0.04	0.01 ±0.03	0.06 ±0.05	0.08 ±0.05	0.17 ±0.06	0.14 ±0.05	0.46 ±0.07	0.44 ±0.07	0.55 ±0.08	0.48 ±0.07	62.17 ±0.98	68.14 ±0.86	47.08 ±1.14
GA+KL	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	24.46 ±1.54	20.72 ±1.50	0.00 ±0.00
NPO	0.25 ±0.06	0.28 ±0.06	0.22 ±0.06	0.20 ±0.05	0.15 ±0.05	0.13 ±0.05	0.12 ±0.05	0.10 ±0.05	0.12 ±0.06	0.09 ±0.05	58.35 ±1.08	58.49 ±1.16	50.81 ±1.02
NPO+KL	0.66 ±0.07	0.68 ±0.06	0.57 ±0.07	0.58 ±0.06	0.64 ±0.06	0.65 ±0.06	0.70 ±0.07	0.66 ±0.06	0.59 ±0.07	0.60 ±0.06	61.18 ±0.96	61.03 ±1.05	52.99 ±0.94
RLabel	0.01 ±0.04	0.01 ±0.03	0.01 ±0.04	0.00 ±0.00	0.01 ±0.04	0.01 ±0.04	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	24.76 ±1.52	19.57 ±1.56	0.00 ±0.00
PISCES	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	25.58 ±1.48	21.58 ±1.46	0.00 ±0.00
ALKN	0.84 ±0.05	0.80 ±0.05	0.78 ±0.06	0.76 ±0.05	0.74 ±0.06	0.69 ±0.06	0.71 ±0.07	0.65 ±0.06	0.60 ±0.07	0.62 ±0.06	62.17 ±0.88	69.54 ±0.42	**57.09** ±0.92
O^3	**0.93** ±0.04	**0.90** ±0.04	**0.82** ±0.05	0.75 ±0.05	0.73 ±0.06	0.68 ±0.06	0.69 ±0.07	0.62 ±0.06	0.64 ±0.07	0.63 ±0.06	–	–	–
FIT (Ours)	0.90 ±0.04	0.88 ±0.04	0.75 ±0.05	0.71 ±0.05	**0.75** ±0.05	**0.72** ±0.05	**0.76** ±0.05	**0.70** ±0.05	**0.73** ±0.05	**0.71** ±0.05	**65.56** ±0.34	**70.01** ±0.01	57.02 ±0.86
Qwen3-14B													
Retain Model	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	79.99 ±0.00	81.33 ±0.00	84.84 ±0.00
GA	0.62 ±0.08	0.68 ±0.07	0.27 ±0.07	0.27 ±0.06	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	45.29 ±1.56	25.14 ±1.62	0.00 ±0.00
GA+GD	0.87 ±0.06	0.85 ±0.06	0.75 ±0.07	0.61 ±0.07	0.64 ±0.08	0.62 ±0.07	0.69 ±0.08	0.65 ±0.07	0.52 ±0.08	0.44 ±0.07	72.35 ±1.04	77.64 ±0.88	69.30 ±1.16
GA+KL	0.51 ±0.07	0.55 ±0.07	0.28 ±0.07	0.29 ±0.06	0.06 ±0.05	0.06 ±0.05	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	68.08 ±1.22	43.49 ±1.46	0.00 ±0.00
NPO	0.69 ±0.08	0.75 ±0.07	0.55 ±0.07	0.58 ±0.07	0.49 ±0.08	0.48 ±0.07	0.50 ±0.08	0.43 ±0.07	0.48 ±0.08	0.40 ±0.07	72.14 ±1.08	72.48 ±1.12	57.85 ±1.36
NPO+KL	0.69 ±0.07	0.74 ±0.07	0.63 ±0.07	0.68 ±0.06	0.64 ±0.07	0.65 ±0.06	0.70 ±0.07	0.63 ±0.06	0.69 ±0.07	0.60 ±0.06	76.52 ±0.82	75.36 ±1.02	73.14 ±0.94
RLabel	0.25 ±0.06	0.29 ±0.06	0.09 ±0.05	0.09 ±0.05	0.05 ±0.05	0.05 ±0.05	0.05 ±0.05	0.04 ±0.04	0.04 ±0.05	0.03 ±0.04	47.49 ±1.52	28.55 ±1.58	0.00 ±0.00
PISCES	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	54.68 ±1.42	35.49 ±1.54	34.59 ±1.50
ALKN	0.84 ±0.06	0.80 ±0.06	0.78 ±0.06	0.79 ±0.06	0.77 ±0.06	0.75 ±0.05	0.75 ±0.06	0.77 ±0.05	0.74 ±0.06	0.71 ±0.05	77.40 ±0.64	**78.26** ±0.72	**74.68** ±0.78
O^3	0.87 ±0.05	**0.91** ±0.04	0.74 ±0.06	0.73 ±0.05	0.70 ±0.06	0.71 ±0.05	0.75 ±0.06	0.73 ±0.05	0.86 ±0.06	0.75 ±0.05	–	–	–
FIT (Ours)	**0.89** ±0.04	0.90 ±0.04	0.75 ±0.05	0.72 ±0.05	0.76 ±0.05	0.72 ±0.05	**0.83** ±0.05	0.73 ±0.05	**0.89** ±0.05	**0.78** ±0.04	**79.15** ±0.54	77.31 ±0.86	74.28 ±0.82
Llama-2-7B-Chat													
Retain Model	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	47.25 ±0.00	58.89 ±0.00	27.29 ±0.00
GA	0.19 ±0.05	0.18 ±0.05	0.07 ±0.03	0.07 ±0.03	0.03 ±0.02	0.07 ±0.03	0.06 ±0.03	0.08 ±0.03	0.02 ±0.01	0.03 ±0.02	27.04 ±1.42	19.41 ±1.55	0.00 ±0.00
GA+GD	0.86 ±0.04	0.87 ±0.04	0.84 ±0.04	0.82 ±0.04	0.59 ±0.06	0.58 ±0.06	0.80 ±0.04	0.74 ±0.05	0.77 ±0.05	0.72 ±0.05	41.00 ±1.03	50.15 ±1.12	19.99 ±1.26
GA+KL	0.62 ±0.06	0.65 ±0.06	0.18 ±0.05	0.16 ±0.04	0.08 ±0.03	0.06 ±0.03	0.08 ±0.03	0.04 ±0.02	0.07 ±0.03	0.03 ±0.02	27.04 ±1.45	20.39 ±1.48	0.00 ±0.00
NPO	0.74 ±0.05	0.76 ±0.05	0.63 ±0.06	0.60 ±0.06	0.42 ±0.07	0.34 ±0.07	0.59 ±0.06	0.44 ±0.07	0.52 ±0.07	0.46 ±0.07	36.56 ±1.18	49.99 ±1.16	5.84 ±1.34
NPO+KL	0.84 ±0.04	0.86 ±0.04	0.89 ±0.03	0.83 ±0.04	0.88 ±0.03	0.89 ±0.03	0.82 ±0.04	0.86 ±0.04	0.89 ±0.03	0.82 ±0.04	42.69 ±0.94	51.03 ±1.05	21.20 ±1.18
RLabel	0.32 ±0.06	0.30 ±0.06	0.22 ±0.05	0.19 ±0.05	0.15 ±0.04	0.13 ±0.04	0.11 ±0.04	0.08 ±0.03	0.09 ±0.03	0.07 ±0.03	23.12 ±1.58	19.57 ±1.56	0.00 ±0.00
PISCES	0.21 ±0.05	0.22 ±0.05	0.15 ±0.04	0.14 ±0.04	0.15 ±0.04	0.16 ±0.04	0.17 ±0.04	0.12 ±0.04	0.15 ±0.04	0.14 ±0.04	24.58 ±1.52	21.58 ±1.44	0.00 ±0.00
ALKN	0.82 ±0.04	0.79 ±0.05	0.78 ±0.05	0.76 ±0.05	0.74 ±0.05	0.74 ±0.05	0.75 ±0.05	0.72 ±0.05	0.76 ±0.05	0.73 ±0.05	45.28 ±0.72	**58.05** ±0.54	24.81 ±0.91
O^3	0.87 ±0.04	0.92 ±0.03	0.84 ±0.04	0.85 ±0.04	0.81 ±0.04	0.87 ±0.04	0.80 ±0.04	0.86 ±0.04	0.64 ±0.06	0.78 ±0.05	–	–	–
FIT (Ours)	**0.92** ±0.03	**0.94** ±0.02	**0.96** ±0.02	**0.98** ±0.01	**0.95** ±0.02	**0.98** ±0.01	**0.93** ±0.03	**0.99** ±0.01	**0.94** ±0.02	**0.98** ±0.01	**47.03** ±0.18	57.83 ±0.62	**26.38** ±0.58
Llama-3-8B-Instruct													
Retain Model	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	67.04 ±0.00	75.27 ±0.00	76.35 ±0.00
GA	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	22.95 ±1.58	19.57 ±1.56	0.00 ±0.00
GA+GD	0.02 ±0.02	0.03 ±0.02	0.15 ±0.04	0.15 ±0.04	0.43 ±0.07	0.43 ±0.07	0.08 ±0.03	0.06 ±0.03	0.24 ±0.05	0.21 ±0.05	24.57 ±1.48	68.74 ±0.98	1.14 ±0.84
GA+KL	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	27.13 ±1.42	19.57 ±1.54	0.00 ±0.00
NPO	0.17 ±0.04	0.12 ±0.04	0.11 ±0.04	0.10 ±0.03	0.10 ±0.03	0.09 ±0.03	0.08 ±0.03	0.07 ±0.03	0.07 ±0.03	0.07 ±0.03	23.12 ±1.55	19.58 ±1.52	0.00 ±0.00
NPO+KL	0.58 ±0.06	0.68 ±0.06	0.59 ±0.06	0.69 ±0.06	0.58 ±0.06	0.66 ±0.06	0.59 ±0.06	0.65 ±0.06	0.61 ±0.06	0.65 ±0.06	65.91 ±0.74	69.56 ±0.92	62.20 ±1.02
RLabel	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	24.46 ±1.50	19.25 ±1.58	0.00 ±0.00
PISCES	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	26.12 ±1.44	22.15 ±1.42	0.00 ±0.00
ALKN	0.81 ±0.04	0.77 ±0.05	0.75 ±0.05	0.76 ±0.05	0.84 ±0.04	**0.80** ±0.04	0.72 ±0.05	0.71 ±0.05	0.71 ±0.05	0.70 ±0.05	58.00 ±1.06	**75.21** ±0.04	60.61 ±1.12
O^3	0.89 ±0.03	**0.88** ±0.03	**0.86** ±0.04	**0.82** ±0.04	**0.88** ±0.03	0.78 ±0.05	0.79 ±0.05	0.72 ±0.05	0.74 ±0.05	0.70 ±0.05	–	–	–
FIT (Ours)	**0.92** ±0.03	0.87 ±0.04	0.85 ±0.04	**0.82** ±0.04	0.87 ±0.04	0.79 ±0.04	**0.80** ±0.04	**0.76** ±0.04	**0.82** ±0.04	**0.78** ±0.04	**66.45** ±0.42	74.57 ±0.54	**65.07** ±0.86
Yi-6B													
Retain Model	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	64.03 ±0.00	73.55 ±0.00	38.36 ±0.00
GA	0.01 ±0.01	0.02 ±0.02	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	27.04 ±1.46	20.56 ±1.50	0.00 ±0.00
GA+GD	0.52 ±0.07	0.62 ±0.06	0.64 ±0.06	0.75 ±0.05	0.62 ±0.06	0.69 ±0.05	0.63 ±0.06	0.63 ±0.06	0.67 ±0.06	0.64 ±0.06	45.99 ±1.12	65.56 ±1.02	17.59 ±1.28
GA+KL	0.18 ±0.04	0.28 ±0.06	0.02 ±0.02	0.01 ±0.01	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	24.46 ±1.52	20.88 ±1.48	0.00 ±0.00
NPO	0.57 ±0.06	0.62 ±0.06	0.43 ±0.07	0.47 ±0.07	0.18 ±0.05	0.17 ±0.04	0.11 ±0.04	0.11 ±0.04	0.04 ±0.02	0.07 ±0.03	24.35 ±1.54	20.64 ±1.50	0.00 ±0.00
NPO+KL	0.78 ±0.05	0.81 ±0.04	0.76 ±0.05	0.74 ±0.05	0.67 ±0.06	0.76 ±0.05	0.68 ±0.06	0.69 ±0.06	0.69 ±0.06	0.69 ±0.06	57.40 ±0.96	66.34 ±0.98	31.91 ±1.08
RLabel	0.16 ±0.04	0.19 ±0.04	0.02 ±0.02	0.02 ±0.02	0.01 ±0.01	0.01 ±0.01	0.01 ±0.01	0.01 ±0.01	0.01 ±0.01	0.01 ±0.01	23.12 ±1.58	19.57 ±1.56	0.00 ±0.00
PISCES	0.12 ±0.04	0.08 ±0.03	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	25.57 ±1.46	20.88 ±1.48	0.00 ±0.00
ALKN	0.87 ±0.04	0.85 ±0.04	0.76 ±0.05	0.77 ±0.05	0.81 ±0.04	0.82 ±0.04	0.77 ±0.05	0.75 ±0.05	0.76 ±0.05	0.74 ±0.05	58.52 ±0.92	66.34 ±0.96	32.14 ±1.04
O^3	0.93 ±0.03	**0.87** ±0.03	0.90 ±0.03	0.82 ±0.04	0.89 ±0.03	0.82 ±0.04	0.85 ±0.04	**0.83** ±0.04	0.89 ±0.03	0.80 ±0.04	–	–	–
FIT (Ours)	**0.95** ±0.02	0.85 ±0.04	**0.94** ±0.02	**0.86** ±0.03	**0.94** ±0.02	**0.85** ±0.03	**0.92** ±0.03	**0.83** ±0.04	**0.91** ±0.03	**0.82** ±0.04	**63.52** ±0.38	**72.90** ±0.52	**37.58** ±0.60

Table 9: Five-stage component ablations on Llama-3-8B (mean ± standard deviation over five runs). Downstream utility is evaluated after 300 requests.

Method	F.D.$^{(60)}$	R.U.$^{(60)}$	F.D.$^{(120)}$	R.U.$^{(120)}$	F.D.$^{(180)}$	R.U.$^{(180)}$	F.D.$^{(240)}$	R.U.$^{(240)}$	F.D.$^{(300)}$	R.U.$^{(300)}$	MMLU	CSQA	GSM8K
Retain Model	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	1.00 ±0.00	65.97 ±0.00	70.02 ±0.00	58.83 ±0.00
w/o adaptive routing	0.79 ±0.05	0.82 ±0.05	0.50 ±0.06	0.49 ±0.05	0.16 ±0.06	0.10 ±0.05	0.20 ±0.06	0.14 ±0.05	0.14 ±0.07	0.07 ±0.06	62.40 ±0.96	63.80 ±1.08	14.12 ±1.42
w/o redundancy filtering	0.65 ±0.07	0.76 ±0.06	0.59 ±0.07	0.63 ±0.06	0.61 ±0.06	0.61 ±0.06	0.64 ±0.07	0.59 ±0.06	0.66 ±0.07	0.63 ±0.06	64.17 ±0.72	68.21 ±0.84	55.72 ±1.02
w/o targeted layers	0.53 ±0.06	0.59 ±0.05	0.43 ±0.07	0.51 ±0.06	0.43 ±0.06	0.51 ±0.05	0.51 ±0.05	0.53 ±0.06	0.40 ±0.07	0.48 ±0.06	64.02 ±0.78	67.12 ±0.92	54.98 ±1.08
FIT (Ours)	**0.90** ±0.04	**0.88** ±0.04	**0.75** ±0.05	**0.71** ±0.05	**0.75** ±0.05	**0.72** ±0.05	**0.76** ±0.05	**0.70** ±0.05	**0.73** ±0.05	**0.71** ±0.05	**65.56** ±0.34	**70.01** ±0.01	**57.02** ±0.86

G.3 Efficiency Analysis

We use peak GPU memory as a system-level efficiency measure. As shown in Table 10, FIT requires 41.84 GB on Llama-2-7B-Chat, substantially less than full-update baselines (79.20-89.07 GB) and ALKN (140.91 GB), but more than adapter-based O^3 (22.14 GB) and PISCES (27.53 GB). This intermediate footprint reflects FIT's design: request-specific layer attribution updates a broader, more recovery-resistant parameter subset than LoRA while avoiding full-model optimization.

Table 10: Peak GPU memory usage on Llama-2-7B-Chat. Lower is better.

Method	GA	GA+GD	GA+KL	NPO	NPO+KL	RLabel	PISCES	O^3	ALKN	FIT
Memory (GB)	81.71	88.43	89.07	81.71	86.72	79.20	27.53	**22.14**	140.91	41.84

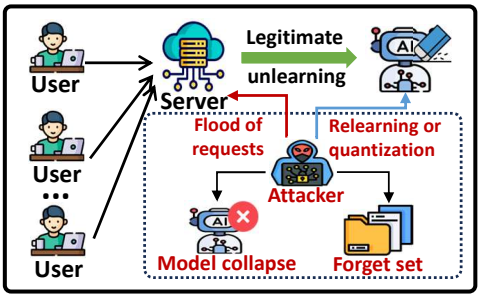


Figure 12: Threat model for continual unlearning: *malicious unlearning* or DoS attacks, *relearning*, and *quantization* attacks considered throughout our security evaluation.

H Threat Model

Attacker's Goal. Figure 12 depicts the interaction among the server, legitimate users, and an adversary under the continual unlearning setting. The server hosts an LLM, answers inference queries, and processes unlearning requests. The adversary masquerades as a normal user and can mount two types of attacks: i) *Malicious unlearning*: A rapid stream of unlearning requests forces repeated updates, inducing catastrophic forgetting and degrading performance far beyond the designated forget set (*cf.* a "denial-of-service" attack on model utility), or ii) *Post-unlearning recovery*: After the unlearned model is deployed, the attacker attempts to restore the erased knowledge through, *e.g.*, *relearning via fine-tuning* (Lo et al., 2024; Barez et al., 2025), or *quantization attacks* that compress the model to low-bit precision, amplifying residual memorization and making it easier to extract sensitive information from the deployed model (Zhang et al., 2025b).

Attacker's Capabilities. Both *malicious unlearning* and *relearning* attacks often assume a *black-box* setting. Malicious unlearning further assumes that the adversary can issue a burst of unlearning requests in a short time window, inducing catastrophic forgetting and model collapse.

In contrast, relearning assumes that an adversary can obtain auxiliary data and fine-tune the unlearned model. Such data may include (i) a subset of the forget set, or in the worst case the entire forget set (Barez et al., 2025; Xu et al., 2026), which we do not consider since the forget set is assumed to be private and not observable to the adversary; (ii) the retain set or other samples drawn from a distribution similar to the forget set; or (iii) completely unrelated, out-of-distribution data. *Quantization attacks* (Zhang et al., 2025b) work in a white-box setting with full access to model parameters so that the attacker can subject the weights to aggressive low-bit quantization. As the weights of the original model and the unlearned model typically differ only slightly, the quantizer rounds both to the same value, nullifying the effect of unlearning, *e.g.*, 2.301235 (original) vs. 2.412567 (unlearned) $\rightarrow$ 2.2858 after `int4`, thus restoring much of the unlearned information due to "weights collision."

Threats that require altering server infrastructure, modifying the unlearning algorithm, or intercepting private communications between the server and its users are out of scope.