TASE: ULTRA-SPARSE VECTOR-LEVEL UNLEARNING VIA TOKEN-ANCHORED SURGICAL EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Current approximate unlearning methods aim to suppress the influence of target information and preserve general capabilities. Achieving both objectives remains challenging since Gradient Ascent incurs substantial training cost and inconsistent retained utility, while parameter-efficient LoRA-based approaches still struggle to achieve strong forgetting without sacrificing retained utility. Building on our observation that sparse edits to only a small set of forget-associated token embeddings suffice to alter target outputs, we propose TASE, a token-anchored approximate unlearning method with extreme parameter efficiency. TASE exploits token distribution differences between the forget and retain sets to construct a small anchored token set that prioritizes tokens concentrated in the forget set but uncommon in retained data. Embedding-only surgical editing then updates only the corresponding embedding rows, achieving targeted knowledge suppression and retained-utility preservation through ultra-sparse vector-level unlearning. The anchored-token sparse update strategy further supports token-based unlearning requests, reducing data exposure in remote model update settings. Experiments on BERT, Phi-1.5B, and Llama-2-7B across classification, TOFU, and MUSE demonstrate competitive forgetting and retained utility across encoder and decoder architectures. TASE modifies 2-5 orders of magnitude smaller parameters than existing approaches, while reducing peak GPU memory and GPU-hours by nearly 30%.

1 INTRODUCTION

Large language models acquire their capabilities from training on massive text corpora, a process that can also cause them to memorize private, copyrighted, or otherwise removable content Kassem et al. (2023); Jang et al. (2023); Yan et al. (2024). Data-protection regulations such as the GDPR establish requirements for data deletion, creating a growing need for trained models to accommodate removal requests after training Voigt & von dem Bussche (2017). Retraining from scratch on the retain set, often regarded as an exact unlearning approach, provides a standard for evaluating unlearning effectiveness, but requires full-model optimization with substantial memory and computational cost Bourtoule et al. (2021); Yan et al. (2022). Rather than rebuilding the model, approximate unlearning reduces the influence of the forget set through parameter editing and is commonly evaluated along three dimensions: unlearning effectiveness, retained utility, and computational cost relative to retraining Maini et al. (2024); Shi et al. (2025); Dorna et al. (2025).

Gradient ascent (GA) reduces the influence of targeted information by increasing the loss on the corresponding forget examples Yuanshun et al. (2023); Maini et al. (2024). However, its unbounded objective continues to drive updates after the targeted information has been sufficiently suppressed, since the optimized parameters are shared with retain inputs. This parameter drift can degrade retained knowledge and destabilize generation Zhang et al. (2024); Cha et al. (2025). Existing methods stabilize unlearning and preserve retained behavior through retain-set regularization or reference-model constraints Kurmanji et al. (2023); Zhang et al. (2024); Dong et al. (2025), yet still optimize over a broad shared parameter space, increasing computational cost and perturbing parameters important to retained knowledge.

LoRA-based unlearning methods Hu et al. (2022); Cha et al. (2025) reduce the number of trainable parameters by replacing full-model optimization with low-rank adapters, thereby lowering memory

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.

and update cost. However, the scope of parameter updates is determined by the adapter structure rather than the forget set. The same adapters process both forget and retain inputs, restricting optimization to a low-rank subspace of the selected modules without localizing updates to parameters associated with the targeted information. Increasing the adapter rank improves the capacity for target suppression, but can also interfere with retained utility, as shown in Figure 1. Moreover, the number of trainable parameters is determined by the adapter rank and module dimensions, scaling with model size while remaining independent of the size of the forget set. Retain examples are also commonly incorporated into optimization to preserve retained utility, making computational cost dependent not only on the number of trainable parameters but also on the amount of data processed and the number of optimization steps. Recent methods improve the granularity of unlearning through instance-wise specification Cha et al. (2024) or token-level importance estimation Wan et al. (2025). However, such selectivity determines only which examples or tokens contribute to the unlearning objective, while optimization is still performed over the full parameter set.

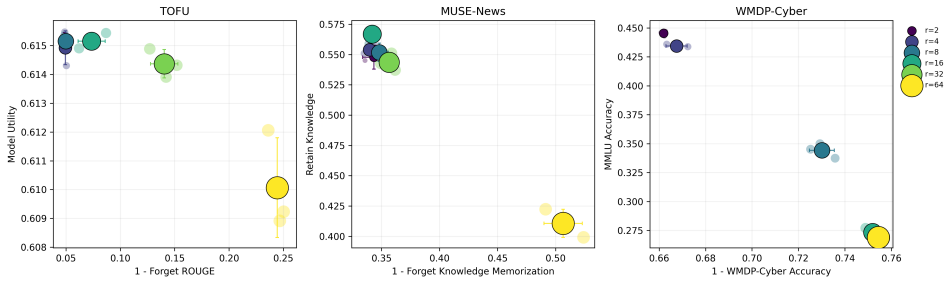


Figure 1: **Motivation of anchored token.** The effects of LoRA rank on parameter efficiency and forget/retain performance are evaluated on three benchmarks. Increasing the rank improves forget performance but degrades retain performance and requires more computational resources.

To reduce the resource cost of unlearning while limiting unnecessary parameter perturbations, we propose TASE, a parameter- and compute-efficient approximate unlearning method enabled by Embedding-Only Surgical Editing. Our empirical analysis shows that restricting the update to a small fraction of forget-associated token embeddings achieves competitive unlearning performance relative to retraining. Motivated by this observation, we design anchored token selectors to identify a small set of tokens that cover the forget set while minimizing their overlap with the retain set, thereby concentrating the unlearning signal on forget-specific information. We then introduce an Ultra-Sparse Vector-Level Unlearning strategy that updates only the embedding vectors indexed by the selected tokens while freezing all remaining parameters. Each row update is normalized prior to application, and the edited vector is rescaled to its pre-unlearning norm, calibrating the update magnitude while limiting radial drift on the selected embeddings. By restricting optimization to these anchor-indexed vectors and requiring neither retain-set optimization nor a reference model, TASE substantially reduces the trainable parameter scope and associated computational cost. Experiments on both encoder- and decoder-based models demonstrate the applicability of TASE across model architectures, while results on Phi-1.5B and Llama2-7B further show its scalability across model sizes and families.

- We introduce anchored token selectors that distinguish forget-specific tokens from broadly shared ones, providing targeted anchors for effective unlearning with limited interference to retained utility.
- We propose an embedding-only surgical editing strategy that confines optimization to the anchored embedding vectors, avoiding broad parameter perturbations and enabling ultra-sparse vector-level unlearning with extreme parameter and computational efficiency.
- We demonstrate that TASE achieves extreme parameter efficiency across encoder and decoder architectures and scales across model sizes and families, while maintaining competitive post-unlearning performance against existing baselines.

2 EMPIRICAL OBSERVATION

Recent studies have explored finer-grained unlearning through instance-wise specification and token-level importance estimation Cha et al. (2024); Wan et al. (2025), indicating that unlearning can be targeted to specific parts of the forget data. This raises the question of whether such token-level selectivity can also be translated into correspondingly localized parameter updates.

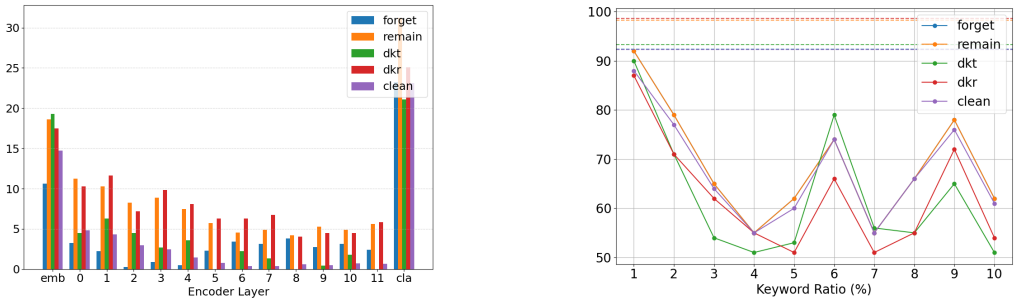


Figure 2: **Observation of anchored token. Left:** Layer-wise unlearning on BERT (IMDb forget-1%) shows that unrestricted embedding-layer updates cause severe degradation in retained utility. **Right:** Updating only 1% of forget-associated token embeddings achieves competitive unlearning performance relative to retraining, while larger update ratios reduce overall utility.

Empirical Analysis. We first investigate whether token embeddings provide sufficient capacity for unlearning through a BERT layer-wise sensitivity analysis. As shown in Figure 2**(Left)**, updating the full embedding layer achieves effective forgetting but substantially degrades retained utility. This indicates that the embedding layer has sufficient capacity to drive forgetting, while indiscriminate optimization of all embedding vectors introduces unnecessary interference with retained information. In contrast, restricting parameter updates to a small fraction of forget-associated token embeddings achieves competitive unlearning performance relative to retraining, as shown in Figure 2**(Right)**. A possible explanation for why such sparse updates can effectively suppress the influence of forget examples is that, although each update is confined to a single embedding vector, its effect propagates through Transformer contextualization and can alter the representation of the entire sequence in which the token occurs. The above results show that the output of target sample can be suppression through extremely efficient sparse updates to a small set of token anchors. Building on this observation, TASE achieves parameter-efficient unlearning while preserving retained utility by addressing two key questions: how to identify tokens that are sufficiently specific or important to the forget set, and how to surgically update the corresponding token embeddings without causing over-forgetting of retained knowledge.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

Given an input context x and a target sequence $y = (y_1, \ldots, y_T)$ of length T, a language model parameterized by θ models the conditional likelihood as $p_\theta(y \mid x) = \prod_{j=1}^{T} p_\theta(y_j \mid x, y_{<j})$, where $y_{<j}$ denotes the target tokens preceding position j. After training, an unlearning request specifies a subset $\mathcal{D}_f \subset \mathcal{D}$, referred to as the forget set, whose influence should be removed from the model. The retain set $\mathcal{D}_r$ refers to data that should be preserved after unlearning. Starting from the original parameters θ, approximate unlearning produces updated parameters $\hat{\theta} = \theta + \Delta\theta$, where $\Delta\theta$ denotes the unlearning-induced parameter change.

3.2 OVERVIEW OF TOKEN-ANCHORED SURGICAL EDITING

Section 2 shows that the parameter changes required for effective forgetting can be concentrated on a small subset of forget-associated token embeddings. TASE incorporates this property by restricting the support of $\Delta\theta$ to selected rows of the token embedding matrix. Let $\mathcal{V}$ denote the vocabulary

and $E \in \mathbb{R}^{|\mathcal{V}| \times d}$ denote the token embedding matrix with embedding dimension d. For each token $t \in \mathcal{V}$, we denote its embedding vector by $e_t \in \mathbb{R}^d$, corresponding to the t-th row of E. Given an anchor set $\mathcal{K} \subseteq \mathcal{V}$, TASE optimizes only the selected embedding vectors $E_{\mathcal{K}} = \{e_t \mid t \in \mathcal{K}\}$, while all remaining embedding vectors and non-embedding parameters remain fixed. With the update restricted to $E_{\mathcal{K}}$, TASE must determine which tokens should form the anchor set $\mathcal{K}$ and how their corresponding embedding vectors should be modified while limiting over-forgetting of retained knowledge. We address these two aspects through *Anchored Token Selection* and *Embedding-Only Surgical Editing*, respectively.

3.3 Anchored Token Selection

As shown in Figure 2, restricting optimization to only 1% of forget-associated token embeddings achieves competitive unlearning performance relative to retraining. When only a small fraction of embedding vectors is exposed to optimization, the choice of token anchors becomes critical because the selected tokens should occur repeatedly in forget examples while remaining comparatively uncommon in retained data. Since the same embedding vector e_t is reused whenever token t occurs, modifying e_t affects every example containing t, and subsequent Transformer contextualization propagates the resulting change beyond the edited token to the representation of the containing sequence. A token that occurs frequently in $\mathcal{D}_r$ therefore exposes more retained examples to the same embedding edit and increases retain-side interference, whereas frequent occurrence in $\mathcal{D}_f$ allows the edit to influence more examples targeted for unlearning. TASE accordingly uses the occurrence frequencies of t in $\mathcal{D}_f$ and $\mathcal{D}_r$ as a proxy for the relative intervention exposure of e_t, prioritizing tokens that occur frequently in the forget set but infrequently in the retain set to localize surgical editing toward the forget information while limiting interference with retained knowledge.

We first filter special tokens, punctuation, and stop words to obtain a candidate vocabulary $\mathcal{V}_c \subseteq \mathcal{V}$. For each $t \in \mathcal{V}_c$, let f_t and r_t denote its occurrence counts in $\mathcal{D}_f$ and $\mathcal{D}_r$, respectively, and let $F = \sum_{t'} f_{t'}$ and $R = \sum_{t'} r_{t'}$ denote the corresponding total token counts. We score each candidate as:

$$s(t) = \frac{f_t/F}{r_t/R + \epsilon} \cdot \rho(f_t), \qquad \rho(f_t) = \min\left(1, \frac{f_t}{\tau_{\text{lf}}}\right), \tag{1}$$

where $\epsilon > 0$ is a smoothing constant. The normalized frequency ratio quantifies the relative concentration of t in the forget set with respect to the retain set. Since a large ratio can be produced by only a few forget-side occurrences, $\rho(f_t)$ attenuates candidates with insufficient forget-side support, where τ_{lf} specifies the support threshold.

TASE selects the K candidates with the largest scores $s(t)$ to form the anchor set $\mathcal{K}$. The retain set contributes only through r_t and R when computing $s(t)$ before optimization. Once $\mathcal{K}$ is fixed, the subsequent embedding editing requires no retain examples, retain-side loss, or retain gradients. Decoupling the editing stage from retain-set optimization avoids repeated forward and backward passes over $\mathcal{D}_r$, eliminates the need to balance retain- and forget-side objectives, and removes the associated storage of retain-side activations and gradients, thereby reducing both computational and memory overhead during model updates.

Encoder setting. For classification tasks using encoder-only models, each forget example is associated with a single supervised label, allowing the classification loss to provide a direct sample-level signal of how strongly each token embedding influences the forget prediction. We therefore use a Fisher-style gradient sensitivity estimate to rank candidate anchors in this setting. Once $\mathcal{K}$ is determined, TASE applies the same embedding-only surgical editing procedure described above. The detailed formulation is provided in the supplementary material.

3.4 Embedding-Only Surgical Editing

Anchored Token Selection determines which embedding vectors are exposed to unlearning, but localization alone does not control how strongly these vectors are modified. Directly applying the forget gradient makes the displacement of each anchor depend on its raw gradient magnitude, which can vary substantially across tokens, while repeated updates can also accumulate changes in embedding norm. TASE therefore combines row-wise gradient normalization with norm preservation to control the modification of the selected embeddings throughout optimization. Given the anchor

set $\mathcal{K}$, TASE updates only the corresponding embedding vectors $E_{\mathcal{K}}$, while all remaining model parameters are frozen. For each forget example $(x, y) \in \mathcal{D}_f$, TASE suppresses the original target sequence without introducing a replacement response by minimizing:

$$\mathcal{J}_f(x, y) = \frac{1}{T} \sum_{j=1}^{T} \log p_\theta(y_j \mid x, y_{<j}), \tag{2}$$

for each $(x, y) \in \mathcal{D}_f$. Minimizing $\mathcal{J}_f$ decreases the autoregressive likelihood of the original response and is equivalent to gradient ascent on its negative log-likelihood. At optimization step m, the gradient of each selected embedding vector is normalized before the update:

$$\tilde{e}_t^{(m+1)} = e_t^{(m)} - \eta \frac{\nabla_{e_t} \mathcal{J}_f}{\|\nabla_{e_t} \mathcal{J}_f\|_2 + \epsilon_g}, \qquad t \in \mathcal{K}, \tag{3}$$

where η is the step size and $\epsilon_g > 0$ ensures numerical stability. Row-wise gradient normalization makes the displacement of each anchor primarily controlled by η, preventing differences in raw gradient magnitude from producing disproportionately large updates to individual embedding vectors. TASE further rescales each updated vector to its pre-unlearning norm:

$$e_t^{(m+1)} = \tilde{e}_t^{(m+1)} \frac{\|e_t^{(0)}\|_2}{\|\tilde{e}_t^{(m+1)}\|_2}, \qquad t \in \mathcal{K}, \tag{4}$$

where $e_t^{(0)}$ denotes the embedding vector before unlearning. Norm preservation limits radial drift accumulated during optimization while retaining the directional modification introduced by the forget gradient. Row-wise gradient normalization controls the per-step displacement of each anchor, while norm preservation prevents accumulated changes in embedding magnitude. The editing stage therefore operates only on forget examples and $E_{\mathcal{K}}$, requiring neither retain-side optimization nor a reference model.

Keyword-aware setting. We also provide TASE_{Ka}, a lightweight configuration of TASE that further restricts the information available during editing by replacing the original forget examples with inputs constructed from the selected token anchors. The same embedding-only update, gradient normalization, and norm preservation are retained. The construction of the optimization inputs is described in Section A.1.2.

Encoder setting. For classification tasks using encoder-only models, TASE retains the same anchor-restricted embedding update, row-wise gradient normalization, and norm preservation. The autoregressive forget objective in Eq. 2 is replaced by an alternative-label classification objective, since the prediction target is a discrete class rather than a generated sequence. The detailed formulation is provided in the supplementary material.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets and Models

Classification tasks. We evaluate TASE with `bert-base-uncased` on IMDb Maas et al. (2011) and SST-2 Socher et al. (2013). For each dataset, 1% of the training instances are randomly selected as the forget set $\mathcal{D}_f$, and the remaining instances form $\mathcal{D}_r$. Following KGA Wang et al. (2023), we construct semantically close subsets $\mathcal{D}_{kr}$ and $\mathcal{D}_{kt}$ from the retain and test sets by k-means clustering to assess collateral effects on neighboring examples. Retraining on $\mathcal{D}_r$ provides the post-unlearning reference.

Generation tasks. We evaluate factual unlearning on TOFU Maini et al. (2024), which contains question–answer pairs about fictitious authors, using the standard forget01, forget05, and forget10 splits corresponding to 1%, 5%, and 10% of the author profiles. The main TOFU comparison uses Llama-2-7B-chat, and Phi-1.5B provides a cross-model evaluation. We further evaluate on MUSE-News Shi et al. (2025), which uses disjoint forget and retain news corpora and complements TOFU's discrete Q&A setting with long-form document unlearning.

Table 1: Comparison of unlearning methods on TOFU. Bold and underlined values denote the best and second-best results.

	forget01 (1%)					**forget05** (5%)					**forget10** (10%)				
Method	FTR↑	F-RG↓	R-RG↑	MU↑	ΔP↓	FTR↑	F-RG↓	R-RG↑	MU↑	ΔP↓	FTR↑	F-RG↓	R-RG↑	MU↑	ΔP↓
Llama-2-7B															
Retrain	0.683	0.399	0.994	0.618	-	0.674	0.396	0.985	0.619	-	0.682	0.399	0.987	0.600	-
GA	0.574	0.425	0.707	0.588	6.738	0.650	0.243	0.353	0.537	6.738	0.575	0.557	0.635	0.627	6.738
GradDiff	0.571	0.422	0.556	0.597	6.738	0.630	0.399	0.433	0.582	6.738	0.693	0.342	0.368	0.526	6.738
NPO	0.672	0.351	0.477	0.568	6.738	0.719	0.241	0.387	0.565	6.738	0.720	0.222	0.398	0.563	6.738
PDU	0.742	0.158	0.560	0.608	6.738	**0.806**	**0.133**	0.464	0.578	6.738	**0.879**	**0.077**	0.646	0.614	6.738
RMU	0.529	0.905	0.870	0.615	1.35e-01	0.524	0.817	0.837	0.607	1.35e-01	0.520	0.823	0.835	0.618	1.35e-01
SimNPO	0.602	0.400	0.575	0.602	6.738	0.657	0.335	0.478	0.581	6.738	0.678	0.347	0.439	0.588	6.738
UNDIAL	0.646	0.332	0.767	**0.645**	6.738	0.659	0.281	0.601	**0.643**	6.738	0.658	0.278	0.632	**0.642**	6.738
TASE	**0.839**	**0.019**	**0.980**	0.609	**8.2e-05**	0.683	0.319	**0.985**	0.614	**2.0e-04**	0.727	0.181	**0.936**	0.605	**2.3e-04**
Phi-1.5B															
Retrain	0.638	0.432	0.624	0.409	-	0.612	0.418	0.646	0.408	-	0.619	0.414	0.639	0.414	-
GA	0.550	0.451	0.512	**0.399**	1.418	0.514	0.528	0.569	**0.407**	1.418	0.631	0.443	0.438	0.326	1.418
GradDiff	0.630	0.229	0.419	0.316	1.418	0.558	0.399	0.516	0.390	1.418	0.595	0.380	0.546	0.384	1.418
NPO	0.628	0.292	0.484	0.386	1.418	0.582	0.410	0.554	0.400	1.418	0.618	0.263	0.544	0.387	1.418
PDU	0.681	**0.181**	0.393	0.342	1.418	**0.655**	**0.263**	0.581	0.388	1.418	**0.751**	**0.204**	0.645	0.406	1.418
RMU	0.492	0.648	0.595	0.383	5.03e-02	0.495	0.488	0.495	0.356	5.03e-02	0.496	0.592	0.611	0.396	5.03e-02
SimNPO	0.610	0.254	0.529	0.374	1.418	0.580	0.385	0.603	0.397	1.418	0.608	0.362	**0.647**	**0.408**	1.418
UNDIAL	**0.704**	0.286	0.440	0.364	1.418	0.621	0.323	0.476	0.382	1.418	0.662	0.321	0.553	0.367	1.418
TASE	0.668	0.307	**0.627**	0.393	**3.3e-05**	0.620	0.275	**0.624**	0.395	**1.1e-04**	0.596	0.247	0.621	0.399	**1.3e-04**

4.1.2 BASELINES

For classification tasks, we compare TASE with Retrain, high-learning-rate fine-tuning (Fin), Gradient Ascent (GA), and KGA Wang et al. (2023). For generation tasks, we report the pre-unlearning Full-SFT model and retain-only Retrain model as references, and compare TASE with GA Yao et al. (2024), GradDiff Maini et al. (2024), NPO Zhang et al. (2024), SimNPO Fan et al. (2025), RMU Li et al. (2024), PDU Entesari et al. (2025), and UNDIAL Dong et al. (2025). All methods start from the same task model within each benchmark; detailed baseline configurations are provided in Appendix A.2.

4.1.3 EVALUATION METRICS

Classification tasks. We report Accuracy and F1-score on $\mathcal{D}_f$, $\mathcal{D}_r$, and $\mathcal{D}_t$ to measure forgetting and retained task utility. Performance on $\mathcal{D}_{kr}$ and $\mathcal{D}_{kt}$ quantifies collateral effects on semantically neighboring examples. We additionally report $\Delta P \downarrow$, the number of scalar parameters included in the update support during unlearning, to measure structural update sparsity. Membership-inference results are reported in Appendix A.3.2.

Generation tasks. On TOFU, Forget Truth Ratio (FTR) ↑ measures the relative likelihood assigned to perturbed versus paraphrased answers, while Forget Q&A ROUGE (F-RG) ↓ measures residual overlap with the target answer. Retain Q&A ROUGE (R-RG) ↑ measures retained factual behavior, and Model Utility (MU) ↑ aggregates general utility over the benchmark evaluation sets. On MUSE-News, Forget Knowledge (FK) ↓ and Forget Verbatim (FV) ↓ measure residual knowledge and verbatim memorization, PrivLeak (PL) measures privacy leakage, and Retain Knowledge (RK) ↑ measures retained corpus knowledge. Across generation benchmarks, ΔP denotes the size of the update support: the full model for full-parameter methods, the optimized target layers for RMU, and $|\mathcal{K}|d$ for TASE. Training time, peak GPU memory, and GPU-hours are reported separately as execution-cost measurements.

4.1.4 IMPLEMENTATION DETAILS

For classification tasks, TASE performs one editing epoch and uses Fisher damping $\lambda = 10^{-3}$ for anchor scoring. For the frequency-based selector, we set $\epsilon = 10^{-4}$ and $\tau_{\text{lf}} = 3$. On Llama-2-7B-chat, (K, η, epochs) is $(20, 0.100, 3)$, $(50, 0.035, 4)$, and $(55, 0.040, 4)$ for forget01, forget05, and forget10, respectively. On Phi-1.5B, we use $(12, 0.10, 3)$ for forget01 and $(48, 0.03, 5)$ for forget05 and forget10. For generation tasks, TASE uses the retain set only for anchor statistics and uses no retain gradients during embedding editing. Additional optimizer, batching, and benchmark-specific settings are provided in Appendix A.2.

4.2 MAIN RESULTS

4.2.1 TOFU RESULTS

Table 1 evaluates whether restricting optimization to anchored embedding rows can balance target suppression and retained utility across models and deletion scales. On Llama-2-7B, TASE achieves the strongest forget01 performance, with FTR $= 0.839$ and F-RG $= 0.019$, while its R-RG of 0.980 remains close to the retrain reference of 0.994. On forget05, TASE exactly matches the retrain R-RG of 0.985 and differs from retraining by only 0.005 in MU. On forget10, it obtains the second-best forget-side results among the unlearning methods while retaining substantially higher R-RG than the full-parameter baselines. PDU provides stronger suppression on the two larger splits, but its broad updates reduce R-RG to 0.464 and 0.646, compared with 0.985 and 0.936 for TASE. This contrast supports the intended role of Anchored Token Selection: tokens concentrated in the forget set provide effective intervention points, while excluding broadly shared tokens limits interference with retained examples.

The same pattern extends to Phi-1.5B. TASE maintains MU within 0.013–0.016 of retraining across all three splits and achieves the highest R-RG on forget01 and forget05, while requiring only 3.3×10^{-5}–1.3×10^{-4} billion supported parameters. Together, the Llama-2-7B and Phi-1.5B results show that the effectiveness of TASE does not arise from a model-specific embedding configuration. As the forget set grows, TASE expands the anchored update support from 8.2×10^{-5} to 2.3×10^{-4} billion parameters on Llama-2-7B, rather than exposing the full model to optimization. This data-dependent expansion explains how the method accommodates larger deletion requests while preserving its ultra-sparse update structure.

4.2.2 MUSE-NEWS RESULTS

Table 2 evaluates document-level unlearning on MUSE-News. TASE attains FK $= 0.009$ and FV $= 0.054$ while achieving the highest RK (0.527) among the compared methods, showing that the method extends beyond TOFU's Q&A format. Its update support contains 6.6×10^{-5} billion parameters, compared with 1.35×10^{-1} billion for RMU and 6.738 billion for full-parameter methods.

Table 2: Comparison of unlearning methods on MUSE-News benchmark.

Method	FK↓	FV↓	PL	RK↑	$\Delta P \downarrow$
GradDiff	0.526	0.420	-95.90	0.461	6.738
NPO	0.581	0.370	-84.88	<u>0.515</u>	6.738
RMU	<u>0.221</u>	0.167	-51.72	0.229	<u>1.35e-01</u>
SimNPO	0.504	0.365	-96.52	0.469	6.738
UNDIAL	0.446	0.176	-98.60	0.372	6.738
PDU	0.438	**0.043**	92.48	0.388	6.738
TASE	**0.009**	<u>0.054</u>	**26.74**	**0.527**	**6.6e-05**

4.2.3 DECODER KEYWORD-ONLY RESULTS

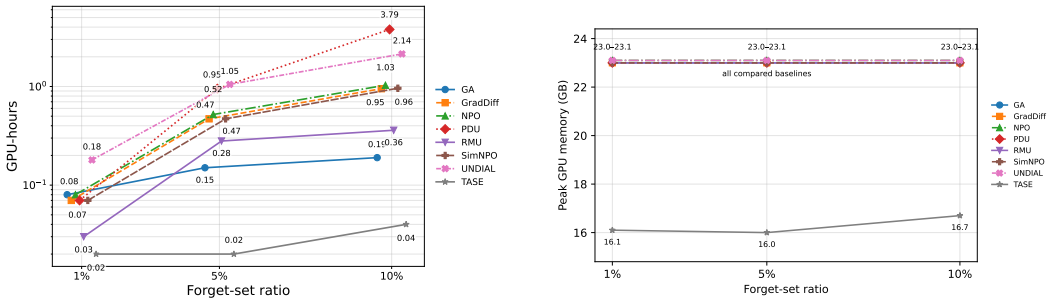
Table 3 compares TASE with its restricted-access decoder variant TASE_{KO} on TOFU. TASE_{KO} performs the editing stage using only token anchors associated with the forget request, without directly updating from the original forget examples. On forget05 and forget10, TASE_{KO} reaches FTR/F-RG of $0.651/0.226$ and $0.674/0.170$ while preserving R-RG of 0.954 and 0.931. On forget01, F-RG remains low (0.042) but FTR drops to 0.013, indicating that the restricted interface does not reproduce the full TASE behavior on the smallest split. Overall, anchor-only editing remains viable on larger deletion requests while exposing a clear limitation at very small forget scales.

Table 3: Decoder keyword-only setting on TOFU.

	forget01					**forget05**					**forget10**				
Method	FTR↑	F-RG↓	R-RG↑	MU↑	$\Delta P \downarrow$	FTR↑	F-RG↓	R-RG↑	MU↑	$\Delta P \downarrow$	FTR↑	F-RG↓	R-RG↑	MU↑	$\Delta P \downarrow$
TASE	0.839	0.019	0.980	0.609	8.2e-05	0.683	0.319	0.985	0.614	2.0e-04	0.727	0.181	0.936	0.605	2.3e-04
TASE_{KO}	0.013	0.042	0.964	0.605	4.9e-05	0.651	0.226	0.954	0.599	2.0e-04	0.674	0.170	0.931	0.601	2.5e-04

4.2.4 COMPUTATIONAL COST

Figure 3 reports GPU-hours and peak GPU memory as the TOFU forget ratio increases from 1% to 10%. TASE uses 0.02, 0.02, and 0.04 GPU-hours across the three splits, while peak memory remains between 16.0 and 16.7 GB. All compared baselines use approximately 23.0–23.1 GB of peak memory, and their GPU-hour cost rises more sharply with forget-set size. Thus, TASE combines a structurally sparse update with lower measured execution cost. The exact measurements are listed in Appendix A.4.1.



(a) GPU-hours.

(b) Peak GPU memory.

Figure 3: Resource scaling on TOFU as the forget-set ratio increases. Markers are slightly offset horizontally within each split to separate overlapping methods.

4.2.5 CLASSIFICATION RESULTS

Table 4 reports the classification results on IMDb and SST-2. TASE maintains high retain and test accuracy while confining the update support to 1.26×10^{-3} billion embedding parameters, compared with 1.09×10^{-1} billion for full-model Fin and GA. On the semantically close subsets, TASE remains near the retrain reference on IMDb and avoids the severe SST-2 degradation observed for KGA on $\mathcal{D}_{kt}$. MIA, keyword-only, and forget-scale analyses are provided in Appendix A.3.

Table 4: Comparison of unlearning methods on IMDb and SST-2 with a 1% forget set.

		Acc						**F1**					
Dataset	**Metric**	Ori	Ret	Fin	GA	KGA	**TASE**	Ori	Ret	Fin	GA	KGA	**TASE**
IMDb	$\mathcal{D}_f$	98.65	91.93	**90.13**	98.21	96.86	96.41	98.65	91.93	**90.09**	98.20	96.85	96.40
	$\mathcal{D}_r$	99.12	98.25	98.13	98.97	99.89	**98.28**	99.12	98.25	98.13	98.97	99.89	**98.28**
	$\mathcal{D}_{kt}$	91.48	93.72	**93.27**	91.48	**93.27**	**93.27**	91.46	93.72	93.26	91.46	**93.27**	93.26
	$\mathcal{D}_{kr}$	99.55	97.76	96.86	99.55	100.00	**96.41**	99.55	97.76	96.86	99.55	**100.00**	97.76
	$\mathcal{D}_t$	92.28	92.48	90.60	92.32	**92.84**	91.88	92.27	92.47	90.59	**92.31**	92.84	91.87
SST-2	$\mathcal{D}_f$	96.20	93.06	**93.39**	95.87	99.83	95.21	96.14	92.92	**93.30**	95.81	99.83	95.14
	$\mathcal{D}_r$	97.20	97.06	96.76	**97.13**	99.96	98.86	97.16	97.02	96.72	**97.10**	99.96	98.84
	$\mathcal{D}_{kt}$	97.19	97.85	97.52	97.02	58.02	**98.51**	96.22	97.06	96.67	96.01	**57.43**	97.97
	$\mathcal{D}_{kr}$	97.69	97.19	95.37	**97.69**	100.00	98.84	97.66	97.15	95.32	**97.66**	100.00	98.83
	$\mathcal{D}_t$	94.46	94.37	93.94	**94.49**	94.52	94.85	94.42	94.30	93.89	**94.45**	**94.45**	94.80
	$\Delta P \downarrow$	—	—	1.09e-01	1.09e-01	1.09e-01	**1.26e-03**						

4.3 ABLATION STUDY

4.3.1 EDITING ABLATION

Table 5 isolates row-wise gradient normalization and norm preservation on TOFU forget01. Without row-wise normalization, FTR falls from 0.839 to 0.531 and F-RG rises from 0.019 to 0.951, showing that the same step size becomes ineffective when raw gradient magnitude determines the displacement. Removing norm preservation has a smaller effect on forgetting but lowers R-RG and MU and produces 4.09% mean and 8.07% maximum row-norm growth. The mean angular drift remains nearly unchanged (from 20.70° to 20.58°. details analysis are provided in Appendix A.4.7), separating the role of norm preservation from directional editing.

Table 5: Core editing ablation on TOFU forget01.

Setting	FTR↑	F-RG↓	R-RG↑	MU↑
TASE	0.839	0.019	0.980	0.609
w/o Row-wise Norm	0.531	0.951	0.986	0.615
w/o Norm Preservation	0.824	0.029	0.967	0.606

Additional analyses, including encoder-side keyword-only evaluation, privacy leakage, recovery robustness, prompt robustness, cross-model behavior, objective compatibility, and causal diagnostics, are provided in Appendix A.

5 Related Work

Gradient-Based and Regularized Unlearning. Early unlearning methods employing gradient ascent (GA) Yao et al. (2024) often suffer from catastrophic forgetting. Subsequent methods stabilize this objective through different mechanisms. GradDiff Maini et al. (2024) adds retain-set regularization, NPO Zhang et al. (2024) and SimNPO Fan et al. (2025) bound preference-style forget objectives, UNDIAL Dong et al. (2025) uses adjusted-logit self-distillation, and PDU Entesari et al. (2025) solves a primal-dual constrained objective. These designs improve the forget-retain tradeoff, but they still optimize broad parameter sets or introduce additional training signals such as retain losses, reference logits, or constraint updates. SU Wan et al. (2025) introduces token-level gradient importance scoring to identify a critical subset of forget-set tokens, performing gradient ascent only on those tokens to reduce collateral damage. While SU narrows which tokens guide the update, it still applies updates across all model parameters.

Parameter-Efficient Unlearning. LoRA and related PEFT techniques freeze the backbone and optimize compact low-rank or adapter parameters Houlsby et al. (2019); Hu et al. (2022); Zaken et al. (2022). Applying this parameterization to unlearning reduces gradient and optimizer-state storage and yields smaller update checkpoints Cha et al. (2025). These savings lower the cost of adaptation, but the adapter remains a transformation shared by forget and retain inputs. TASE addresses this functional limitation by editing vocabulary rows indexed by forget-associated tokens. The resulting update is both compact in the original parameter space and directly tied to identifiable input units, without adding an inference-time module.

Representation-Level Unlearning. Another line of work directly intervenes in the model's representation space. RMU Li et al. (2024) steers hidden states toward random orthogonal vectors, while PISCES Gur-Arieh et al. (2025) and UIPE Wang et al. (2025) suppress targeted knowledge representations in intermediate layers. SalUn Fan et al. (2024) identifies salient parameters via weight saliency for selective modification. However, existing representation-level methods modify intermediate hidden states, which are shared across all inputs and thus risk collateral damage. The input embedding layer (used in our method), by contrast, maps directly to discrete token identities, enabling more surgical interventions.

6 Conclusion

TASE shows that effective approximate unlearning can be achieved by localizing both the unlearning signal and the parameter update to a small set of token embeddings. Across encoder and decoder architectures, TASE maintains high retained utility while achieving competitive unlearning performance under extreme parameter efficiency, with an update support two to five orders of magnitude smaller than those of the compared methods. Anchored Token Selection uses forget–retain token statistics to localize the edit, while row-wise gradient normalization and norm preservation regulate the displacement of the selected embeddings during optimization. The measured reductions in peak GPU memory and GPU-hours show that the sparse update structure also translates into lower execution cost. Results on TOFU, MUSE-News, and multiple model families indicate that token-indexed embeddings provide a practical interface for resource-efficient unlearning without retain-side optimization during the editing stage.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing and to assist with the retrieval and discovery of related literature. All AI-assisted text was reviewed and revised by the authors, and literature identified with AI assistance was independently verified against the original sources before being cited. We take responsibility for the final content of this work, including all text, claims, and citations produced with the aid of generative AI.

ETHICS STATEMENT

Machine unlearning can support privacy, data-deletion, and model-governance requirements by reducing the influence of specified training information after model deployment. The same capability can also be used to selectively alter model behavior, so evaluation should distinguish requested removal from broader degradation of retained knowledge. Our experiments use established research benchmarks and do not involve human-subject or animal experimentation. We report retained-utility and privacy-oriented evaluations alongside forgetting metrics to characterize the effects of the proposed edits beyond target suppression.

REPRODUCIBILITY STATEMENT

The main paper specifies the anchor-selection rule, forget objective, normalized embedding update, norm-preserving projection, datasets, baselines, and primary evaluation metrics. Appendix A provides the encoder-specific formulation, keyword-only construction, benchmark-specific hyperparameters, additional model results, and supplementary ablations.

REFERENCES

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

Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11186–11194, 2024.

Sungmin Cha, Sungjun Cho, Dasol Hwang, and Moontae Lee. Towards robust and parameter-efficient knowledge unlearning for LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=1ExfUpmIW4.

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 *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the ACL*, pp. 8827–8840, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.444. URL https://aclanthology.org/2025.naacl-long.444/.

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, J. Zico Kolter, Zachary C. Lipton, and Pratyush Maini. OpenUnlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics. In *Advances in Neural Information Processing Systems*, volume 38, pp. 48903–48940, 2025. doi: 10.52202/085713-1458. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3e4a38f228427ab819ba7899003a44b1-Abstract-Datasets_and_Benchmarks_Track.html.

Taha Entesari, Arman Hatami, Rinat Khaziev, Anil Ramakrishna, and Mahyar Fazlyab. Constrained entropic unlearning: A primal-dual framework for large language models. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=ZtB34bQI54.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Dennis Wei, Eric Wong, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=B03cITnirg.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 1776–1803, 2025. doi: 10.52202/085713-0058. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/02443e4e008231e0af0855f3cc70ed17-Abstract-Conference.html.

Yoav Gur-Arieh, Clara Haya Suslik, Yihuai Hong, Fazl Barez, and Mor Geva. Precise in-parameter concept erasure in large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 18986–19006, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.960. URL https://aclanthology.org/2025.emnlp-main.960/.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

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 *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.805. URL https://aclanthology.org/2023.acl-long.805.

Aly Kassem, Omar Mahmoud, and Sherif Saad. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4360–4379, 2023.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates Inc., 2023.

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 Bo 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 Alfred 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 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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28525–28550. PMLR, 2024. URL https://proceedings.mlr.press/v235/li24bc.html.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, 2011.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=uK6RPFBMdM.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TArmA033BU.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013.

Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing, Cham, 2017. ISBN 978-3-319-57959-7. doi: 10.1007/978-3-319-57959-7. URL https://doi.org/10.1007/978-3-319-57959-7.

Yixin Wan, Anil Ramakrishna, Kai-Wei Chang, Volkan Cevher, and Rahul Gupta. Not every token needs forgetting: Selective unlearning balancing forgetting and utility in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 1827–1835, 2025.

Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. KGA: A general machine unlearning framework based on knowledge gap alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13264–13276, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.740. URL https://aclanthology.org/2023.acl-long.740.

Wenyu Wang, Mengqi Zhang, Xiaotian Ye, Zhaochun Ren, Pengjie Ren, and Zhumin Chen. UIPE: Enhancing LLM unlearning by removing knowledge related to forgetting targets. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 25212–25227, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1374. URL https://aclanthology.org/2025.findings-emnlp.1374/.

Biwei Yan, Kun Li, Minghui Xu, Yueyan Dong, Yue Zhang, Zhaochun Ren, and Xiuzheng Cheng. On protecting the data privacy of large language models (llms): A survey. *arXiv preprint arXiv:2403.05156*, 2024.

Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. Arcane: An efficient architecture for exact machine unlearning. In *IJCAI*, volume 6, pp. 19, 2022.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8403–8419, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.457. URL https://aclanthology.org/2024.acl-long.457/.

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

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=MXLBXjQkmb.

A Appendix

A.1 Additional Method Details

A.1.1 Encoder Anchor Scoring and Editing Objective

For classification tasks using encoder-only models, TASE scores token anchors from the sensitivity of the supervised classification loss to the corresponding token embeddings. For a token t, let $g_t = \nabla_{e_t} \mathcal{L}_{\text{cls}}$. We use the Fisher-style score

$$s_{\text{enc}}(t) = \left\| g_t \odot \frac{1}{g_t^2 + \lambda} \right\|_2, \tag{5}$$

where $\lambda > 0$ is a damping constant and the squared gradient provides a diagonal empirical-Fisher approximation. Subword tokens are grouped after detokenization, and the highest-scoring candidates determine $\mathcal{K}$. During editing, an alternative class $\hat{y} \neq y$ is sampled from the label set and TASE minimizes

$$\mathcal{J}_f^{\text{enc}}(x, y) = -\log p_\theta(\hat{y} \mid x). \tag{6}$$

The same row-wise gradient normalization and norm-preserving projection in Eqs. 3–4 are then applied only to $E_{\mathcal{K}}$.

A.1.2 Keyword-Only Setting

For classification tasks, TASE_{KO} constructs a pseudo-sample by inserting each selected anchor into an unrelated text segment of approximately the same length as the original forget samples. If $\mathcal{C}$ denotes the unrelated corpus and ℓ_f the average forget-sample length, an anchor k_i produces $x_i^* = \text{Insert}(k_i, \text{Sample}(\mathcal{C}, \ell_f))$. The alternative-label objective is then applied to the pseudo-sample, while the editable parameter set remains restricted to the selected embeddings. For TOFU, the restricted interface uses prompt-side token anchors as the editing input and does not expose the original answer text to the update stage. The parameter-editing rule is otherwise unchanged.

For decoder models, we additionally consider a Keyword-aware mode in which the client transmits only the selected token IDs together with the identifiers of the corresponding forget samples. The server retrieves these samples from its local dataset and computes the forget objective using the first 32 target tokens, while gradients are retained only for the embedding rows indexed by the transmitted token IDs. The same row-wise gradient normalization and norm preservation are then applied to these rows, without using retain gradients. We denote this setting as TASE_{Ka}.

A.2 Additional Experimental Setup

A.2.1 Classification Tasks

All classification experiments use `bert-base-uncased` as the task model. TASE uses Adam with learning rate 2×10^{-2} and one editing epoch, with batch sizes 32 on IMDb and 1024 on SST-2. The Fisher damping coefficient is $\lambda = 10^{-3}$. Retrain uses learning rate 2×10^{-5} for three epochs on $\mathcal{D}_r$, Fin uses 1×10^{-4} for three epochs, and GA uses 1×10^{-6} for three epochs. KGA follows its reported default configuration with learning rate 2×10^{-5}, three epochs, and auxiliary-set size $|\mathcal{D}_n| = 100$.

A.2.2 Generation Tasks

The Llama-2-7B-chat TOFU checkpoints in the main table use batch-level editing with $B = 16$, frequency-ratio anchor selection, no retain gradient, and norm preservation. The final TASE configurations are $(K, \eta, \text{epochs}) = (20, 0.100, 3)$ for forget01, $(50, 0.035, 4)$ for forget05, and $(55, 0.040, 4)$ for forget10. The Phi-1.5B configurations are $(12, 0.10, 3)$ for forget01 and $(48, 0.03, 5)$ for forget05 and forget10. For the frequency selector, $\epsilon = 10^{-4}$ and $\tau_{\text{lf}} = 3$. The MUSE-News paper configuration uses $K = 16$, step size 30, and one editing epoch on Llama-2-7B. Baseline implementations follow their published objectives and the benchmark evaluation framework used for the corresponding main-table runs.

A.3 Additional Classification Results

A.3.1 Keyword-aware Evaluation

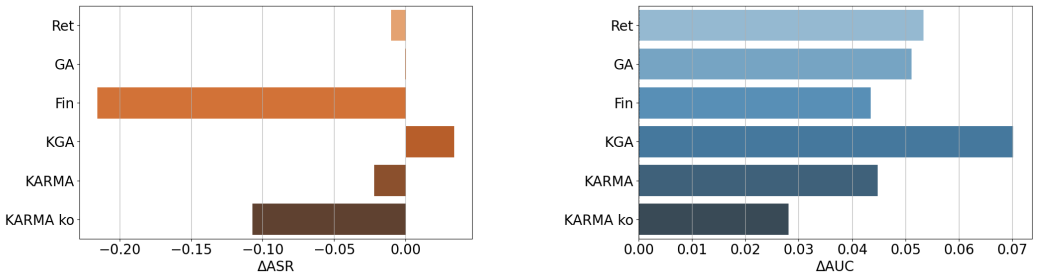
Table 6 compares the encoder-side full-sample and TASE_{Ka} settings. TASE_{KO} remains close to the full-sample variant on IMDb, but the SST-2 retain and test metrics decline more substantially, indicating that the restricted interface is less stable in the encoder setting.

Table 6: Keyword-only setting on classification tasks with BERT.

		IMDb					SST-2				
Method	Metric	$\mathcal{D}_f$	$\mathcal{D}_r$	$\mathcal{D}_{kt}$	$\mathcal{D}_{kr}$	$\mathcal{D}_t$	$\mathcal{D}_f$	$\mathcal{D}_r$	$\mathcal{D}_{kt}$	$\mathcal{D}_{kr}$	$\mathcal{D}_t$
TASE	Acc	95.52	98.26	90.58	96.41	91.88	94.38	96.54	97.36	97.52	95.92
TASE	F1	95.51	90.57	90.58	96.41	91.88	94.34	96.51	96.49	97.49	95.89
TASE_{Ka}	Acc	95.96	97.91	92.38	95.96	91.40	92.40	93.16	95.87	92.40	92.11
TASE_{Ka}	F1	95.96	97.90	91.93	96.85	91.36	92.36	92.46	92.49	92.54	91.76

A.3.2 Membership Inference Attack

We evaluate membership inference with ΔASR and the deviation of attack AUC from random guessing. Figure 4a and Figure 4b compare TASE, TASE_{KO}, and the encoder baselines under the same attack protocol.



(a) ΔASR.

(b) ΔAUC from random guessing.

Figure 4: Membership-inference evaluation for the classification experiments. Lower values indicate lower attack advantage under the reported definitions.

A.3.3 Effect of Forget-Set Size

Table 7 varies the IMDb forget-set ratio from 1% to 30%.

Table 7: TASE accuracy under different forget-set ratios on IMDb.

Metric	1%	5%	10%	15%	20%	30%
$\mathcal{D}_f$	92.38	95.46	94.71	94.99	93.29	88.83
$\mathcal{D}_t$	92.32	92.12	92.04	91.84	91.76	86.47

Test accuracy remains within 0.3 points of the 1% setting through a 10% forget ratio, then declines to 86.47% at 30%, showing that large deletion requests increasingly affect retained generalization.

A.4 Additional Decoder Results

A.4.1 Full Resource Measurements

Table 8 lists the exact forget05 and forget10 measurements underlying Figure 3. TASE remains at 16.0–16.7 GB peak memory and 0.02–0.04 GPU-hours, whereas the compared baselines remain near 23 GB and incur higher GPU-hour cost as the forget set grows.

Table 8: Resource cost on TOFU `forget05` and `forget10`.

Method	Time (h)↓	Mem (GB)↓	GPU Hrs↓
forget05			
GA	0.04	23.0	0.15
GradDiff	0.12	23.0	0.47
NPO	0.14	23.1	0.52
PDU	0.24	23.0	0.95
RMU	0.07	23.0	0.28
SimNPO	0.12	23.0	0.47
UNDIAL	0.27	23.1	1.05
TASE	**0.01**	**16.0**	**0.02**
forget10			
GA	0.06	23.0	0.19
GradDiff	0.25	23.0	0.95
NPO	0.26	23.1	1.03
PDU	0.95	23.0	3.79
RMU	0.10	23.0	0.36
SimNPO	0.25	23.0	0.96
UNDIAL	0.54	23.1	2.14
TASE	**0.01**	**16.7**	**0.04**

A.4.2 Recovery-Based Extraction Robustness

Table 9 evaluates the three main Llama-2-7B checkpoints under direct, noise-prefixed, and composite extraction prompts. The reported recovery rates use a single custom recovery protocol and serve as a supplementary behavioral robustness diagnostic rather than the official PoRT metric.

Table 9: Recovery-based extraction robustness for the main TASE TOFU checkpoints. Lower is better for all reported metrics.

Split	Original↓	Noise↓	Composite↓	Ans. Prob.↓	ROUGE-L↓
forget01	0.0000	0.0000	0.0000	0.0050	0.0337
forget05	0.1850	0.0631	0.1400	0.2712	0.2720
forget10	0.1025	0.0333	0.0450	0.1462	0.1565

No target recovery is observed for forget01 under any of the three attacks. For forget05 and forget10, recovery remains below 0.19 under direct extraction and decreases further under noise-prefix and composite attacks, providing complementary evidence that target answers are not trivially recoverable from the edited checkpoints.

A.4.3 Anchor Composition and Coverage

Table 10 quantifies the coverage and composition of the selected decoder anchors. The selected units are predominantly forget-specific lexical or token-level anchors, not uniformly human-readable semantic keywords.

Table 10: Coverage of selected decoder anchors on TOFU.

Split	#Anchors	Forget coverage	Retain exposure
forget01	12	40/40 (100%)	0 occurrences
forget10	51	380/400 (95.0%)	36/400 examples (9.0%)

On forget01, 50.0% of the selected anchors are opaque BPE or name fragments, 33.3% are proper-name fragments, and 16.7% are broader descriptors. On forget10, the corresponding proportions are 51.0%, 43.1%, and 5.9%, respectively. Roughly half of the selected units in both splits are opaque

BPE or name fragments, confirming that *forget-specific lexical/token-level anchors* is more precise than *semantic keywords*.

A.4.4 PROMPT ROBUSTNESS

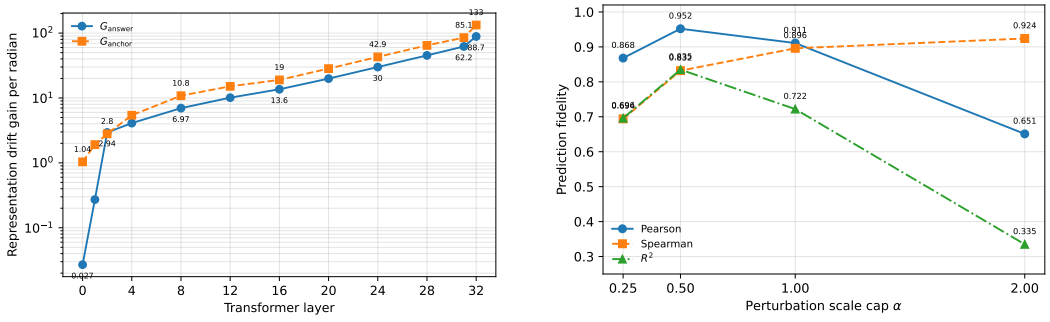
Table 11 evaluates prompt variants beyond the canonical TOFU template. Answer probability remains below 1.8×10^{-3} across all tested variants, but adversarial, keyword-free, and compositional prompts yield substantially higher ROUGE, identifying residual lexical overlap as the harder failure mode.

Table 11: Prompt-robustness stress tests for decoder-side forgetting. Lower is better for both columns.

Stress test	Forget Prob.↓	Forget ROUGE↓
paraphrase	1.07e-6	0.0015
perturbed	1.80e-3	0.0006
alias	8.98e-6	0.0139
multilingual	4.57e-6	0.0025
adversarial extraction	9.13e-5	0.1865
keyword-free	6.22e-5	0.2147
compositional reasoning	9.24e-5	0.2144
highly distributed concept	9.88e-5	0.0841

A.4.5 CAUSAL PATHWAY DIAGNOSTICS

Figure 5 examines how a localized embedding edit propagates through the decoder on TOFU forget01. The answer-region drift gain rises from 0.027 at the embedding output to 88.66 at layer 32, while the anchor-region gain rises from 1.037 to 133.23, showing that the initial row-level perturbation is progressively amplified through contextualization. The local first-order approximation is accurate at moderate perturbation scale ($\alpha = 0.5$: Pearson $= 0.952$, $R^2 = 0.835$) but degrades for larger edits ($R^2 = 0.335$ at $\alpha = 2.0$). Together, the two diagnostics connect a localized embedding perturbation to downstream representation drift and forget-side likelihood change, while also showing where the local linear approximation ceases to be reliable.



(a) Layerwise propagation of anchor perturbations.

(b) Local first-order prediction fidelity.

Figure 5: Causal-pathway diagnostics for anchor-restricted embedding editing on TOFU forget01.

A.4.6 TOFU FORGET QUALITY

Forget Quality (FQ) compares the Truth Ratio distribution of the unlearned model with the retain-only retrain reference using a Kolmogorov–Smirnov test. Table 12 reports FQ for the same TASE checkpoints used in the Llama-2-7B main table.

Table 12: Forget Quality of the main TASE checkpoints on TOFU.

Split	Forget Quality↑
forget01	2.1564×10^{-5}
forget05	0.270474
forget10	0.007945

A.4.7 Editing-Mechanism Diagnostics

Generation diagnostics further separate the two editing controls. Without row-gradient normalization, 39 of 40 forget01 probes still reproduce the correct answer, confirming that the raw-gradient update is ineffective at the same step size. Without norm preservation, edited-row norms grow by 4.09% on average and 8.07% at most, while mean angular drift changes only from 20.70° to 20.58°. These measurements align with the main ablation: normalization controls displacement scale, whereas norm preservation removes radial drift.

A.4.8 Compatibility with a Bounded Forget Objective

We replace the decoder forget objective with SimNPO while retaining the same frequency-ratio selector, 20-row update support, and no retain gradient on forget01, testing whether TASE is compatible with a bounded forget objective.

Table 13: SimNPO objective combined with TASE on TOFU forget01.

Method	FTR↑	F-RG↓	R-RG↑	MU↑	FQ↑
SimNPO, unified rerun	0.563	0.167	0.695	0.588	0.097075
SimNPO–TASE ($\beta = 3.5$)	0.816	0.047	0.974	0.615	0.000188
SimNPO–TASE ($\beta = 4.5$)	0.816	0.038	0.978	0.617	0.001271

Both SimNPO–TASE variants substantially improve FTR, F-RG, R-RG, and MU over the unified SimNPO rerun, showing that the sparse update mechanism is compatible with a bounded forget objective; FQ, however, remains low for these operating points.

A.4.9 General Utility on MMLU

Table 14 evaluates general utility outside TOFU. TASE reaches MMLU accuracies of 0.4579, 0.4593, and 0.4522 across forget01/05/10, remaining within 0.013 of the corresponding retrain references.

Table 14: MMLU accuracy for the main Llama-2-7B-chat TASE checkpoints and retain-only retrain references.

Method	forget01	forget05	forget10
Retrain	0.463	0.465	0.465
TASE	0.4579	0.4593	0.4522

A.5 Cross-Model Decoder Results

A.5.1 Qwen3-8B

Table 15 extends the evaluation to Qwen3-8B using Full-SFT, retain-only retraining, and TASE. TASE reduces F-RG below retraining on all three splits while keeping MU within 0.016 of retraining; R-RG drops more noticeably on forget10, marking the main degradation at the largest deletion scale.

Table 15: Qwen3-8B TOFU extension.

Split	Variant	FTR↑	F-RG↓	R-RG↑	MU↑	Forget Prob.↓	Exact Mem.↓
forget01	Full-SFT	0.411	0.942	0.957	0.653	0.9704	0.9951
forget01	Retrain	0.555	0.405	0.961	0.667	0.0870	0.6582
forget01	TASE	0.567	0.302	0.950	0.653	0.0227	0.6100
forget05	Full-SFT	0.410	0.952	0.959	0.654	0.9721	0.9959
forget05	Retrain	0.532	0.381	0.968	0.662	0.0597	0.6024
forget05	TASE	0.533	0.321	0.952	0.654	0.0338	0.6112
forget10	Full-SFT	0.412	0.939	0.958	0.654	0.9704	0.9951
forget10	Retrain	0.544	0.374	0.982	0.651	0.0572	0.6020
forget10	TASE	0.506	0.324	0.879	0.635	0.0728	0.6076

The forget05 and forget10 configurations use $K = 60$ with $\eta = 0.080$ and 0.100, respectively, for four epochs. Their update-support sizes are 245,005 and 245,266 scalar parameters; forget01 uses 49,368 supported parameters.

A.5.2 WMDP-CYBER PROBE

Table 16 provides a supplementary WMDP-Cyber probe outside TOFU and MUSE. TASE reduces WMDP-Cyber accuracy from 0.3478 to 0.2949 while MMLU decreases from 0.4630 to 0.4547; the result covers only the Cyber subset rather than the full WMDP aggregate.

Table 16: WMDP-Cyber and MMLU supplementary evaluation.

Model	WMDP-Cyber Acc.↓	MMLU Acc.↑
Base Llama-2-7B-chat	0.3478	0.4630
TASE ($K = 60$, $\eta = 0.040$, 80 steps)	0.2949	0.4547