# Stable Forgetting: Bounded Parameter-Efficient Unlearning in Foundation Models

Arpit Garg*, Hemanth Saratchandran*, Ravi Garg, and Simon Lucey

Australian Institute for Machine Learning (AIML),
Adelaide University, Adelaide, Australia
arpit.garg@adelaide.edu.au

**Abstract.** Machine unlearning in foundation models (e.g., language and vision transformers) is essential for privacy and safety; however, existing approaches are unstable and unreliable. A widely used strategy, the gradient difference method, applies gradient descent to retained data while performing gradient ascent on forgotten data. When combined with cross-entropy, this procedure can trigger the unbounded growth of weights and gradients, degrading both forgetting and retention. We provide a theoretical framework that explains this failure by showing how ascent destabilizes optimization in transformer feedforward MLP layers. Guided by this insight, we propose *Bounded Parameter-Efficient Unlearning*, which stabilizes LoRA-based fine-tuning by applying bounded functions to MLP adapters. This controls the weight dynamics during ascent and enables reliable convergence. We validate the approach on Vision Transformer class deletion on CIFAR-100, where GD+Sine is the only evaluated method to achieve both high forget quality and model utility across ViT-B/16, ViT-L/14, and DeiT-S architectures, and demonstrate generality on language-model benchmarks (TOFU, TDEC, MUSE) across architectures from 22M to 8B parameters, achieving improved forgetting while preserving utility.[1]

**Keywords:** Machine Unlearning · Safety and Privacy · Data Governance

## 1 Introduction

The advent of foundation models has profoundly reshaped machine learning; however, their large-scale deployment has revealed critical vulnerabilities in safety and data governance [14]. During pretraining, these models absorb massive datasets that frequently contain sensitive, copyrighted, or personally identifiable information [50], making the ability to selectively *forget* such information a regulatory requirement and a technical necessity [4, 5]. Machine unlearning, which involves removing the influence of specific data without full retraining, is one of the most urgent challenges in the ethical deployment of large-scale models.

* Equal contribution.
[1] Code will be open-sourced upon acceptance.

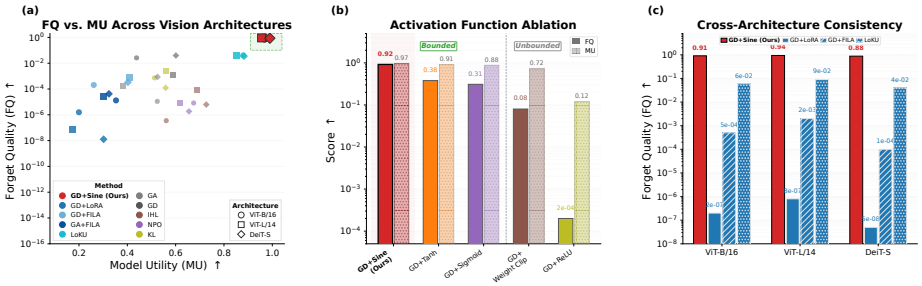


**Fig. 1: GD+Sine is the only method to achieve both high Forget Quality and Model Utility across vision architectures.** **(a)** FQ vs. MU on ViT-B/16, ViT-L/14, and DeiT-S: only GD+Sine (red) reaches the ideal zone (green box), while parameter-efficient and full fine-tuning baselines fail on one or both axes. **(b)** Activation ablation on ViT-B/16: Sine alone achieves near-perfect FQ (0.92) and MU (0.97); unbounded activations collapse on both metrics. **(c)** GD+Sine dominates consistently across all three architectures by orders of magnitude in forget quality over parameter-efficient baselines maintaining model utility.

Current approaches face fundamental limitations in this regard. While the broader literature spans gradient-based, parameter-efficient, preference-based, and representation-editing families (see Sec. 2), the two most widely adopted strategies are as follows. The first is *full fine-tuning*: the standard procedure applies gradient ascent on a forget set with cross-entropy loss [33], leading to instability in training and degradation in retention quality. The *gradient difference* (GD) method [7, 33] addresses this by jointly applying gradient descent on a retention set and gradient ascent on the forget set; however, this combination remains unstable under cross-entropy. The second is *parameter-efficient fine-tuning*, such as LoRA [17], which reduces computational and memory costs but continues to suffer the same instability under gradient difference and cross-entropy across transformer architectures. These limitations have motivated incremental refinements: Fisher information-weighted initialization (FILA) [24] and Inverted Hinge Loss (IHL) [7] improve stability through carefully designed objectives and initialization strategies; however, their combination [7] provides only partial relief. More recent adversarial unlearning frameworks [49] and residual feature alignment methods [25] are fundamentally constrained by their linear parameterization.

We provide a theoretical framework for analyzing the training instability of the gradient difference method under cross-entropy. Our analysis shows that the ascent step causes the weights and gradients in the transformer feedforward MLP (FFN) layers to grow excessively, establishing the root cause. Parameterizing FFN weights with a *bounded function* provides a principled mechanism for stabilizing the optimization under gradient ascent. Building on this, we extended the gradient difference framework with LoRA-based fine-tuning, demonstrating that bounded parameterization directly stabilizes weights and gradients during

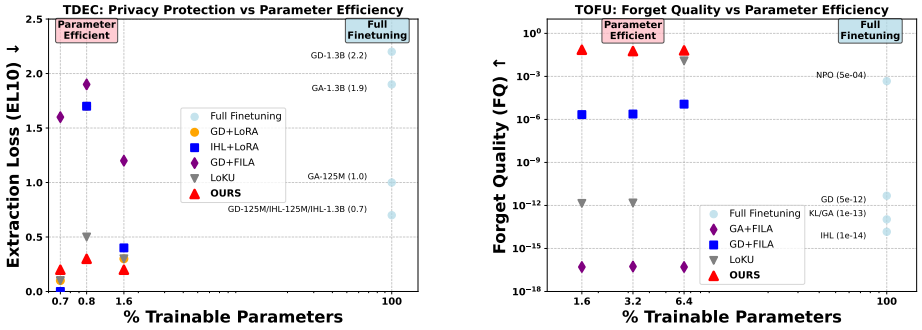


**Fig. 2: Balancing efficiency and effectiveness in parameter tuning.** (**Left**) On TDEC, our method achieves stronger privacy protection than existing parameter-efficient baselines while requiring fewer parameters than full-tuning. (**Right**) On TOFU, our approach maintains consistently high forget quality across LoRA ranks, outperforming state-of-the-art baselines by orders of magnitude while preserving parameter efficiency.

unlearning. We propose **bounded parameter-efficient unlearning**, which applies bounded functions to LoRA adapters in FFN layers, enabling stable fine-tuning with a cross-entropy forgetting objective. This directly overcomes the key limitations of previous methods, providing theoretical guarantees and effectiveness.

We evaluated our framework on standard techniques in the field, including ViT class deletion on CIFAR-100 (Fig. 1), and TOFU, TDEC, and MUSE benchmarks (Fig. 2), spanning ViT, DeiT, GPT-Neo, Phi, and LLaMA architectures (22M-8B parameters). Our method achieves strong unlearning performance with significant improvements in forget quality over existing methods, while preserving model utility. Our main contributions are as follows.

1. We develop a theoretical framework for the gradient difference method with cross-entropy loss, showing instability arises because the ascent step drives uncontrolled growth of weights and gradients in MLP feedforward layers. We derived the key insight that parameterizing the feedforward weights with a bounded function stabilizes the gradient ascent.
2. Building on this principle, we propose **bounded parameter-efficient unlearning**, a parameter-efficient method that applies bounded functions to LoRA adapters in feedforward layers. Our approach achieves stable unlearning, delivering substantial improvements in forget quality while preserving the utility across benchmarks.

## 2 Related Work

*Machine unlearning.* Machine unlearning removes the influence of specific data without full retraining [4, 5]. For foundation models, retraining is infeasible at

scale [35, 40], motivating the development of efficient alternatives. Recent methods can be classified into four categories: (1) *Full fine-tuning* applies gradient-based forgetting objectives [18, 33, 55]. (2) *Parameter-efficient fine-tuning* using LoRA-style adapters [7, 17, 24] (3) *Preference-based methods* that leverage alignment signals [41, 57]; and (4) *representation/weight-editing methods* that directly alter internal activations or weights [19, 34]. Vision approaches target class/concept removal via saliency masking [11], contrastive mechanisms, and parameter-efficient parameterizations [3, 23, 25, 43, 51]. We focus on (1) and (2) as they are the most relevant; our method belongs to category (2). An in-depth survey of each category is presented in Sec. A.

*Full fine-tuning methods.* Applying cross-entropy gradient ascent to the forget set [33] routinely destabilizes training and degrades retention. The *gradient difference* (GD) method [7, 33] adds a gradient descent retention objective to counteract this, yet remains unstable under cross-entropy ascent. FILA [24] and Inverted Hinge Loss (IHL) [7] partially mitigate instability, but neither identifies or resolves the *root cause*: unbounded weight growth. Full fine-tuning is also computationally expensive because it updates all model weights.

*Parameter-efficient fine-tuning.* LoRA [17] factorizes weight updates into low-rank matrices, reducing the overhead from $dk$ to $(d+k)r$. Combined with the gradient difference, LoRA-based unlearning still suffers from severe cross-entropy instability [7, 18]. Recent efforts have addressed this issue via influence reweighting (GUARD [36], RapidUn [58]), gradient reconstruction (R2F [30]), and bootstrapping (LUNE [29], BB [26]), targeting symptoms rather than root causes. As shown in Sec. 3, the fundamental issue is *unbounded gradient ascent dynamics in MLP feedforward layers*; constraining adapter weights solves this and complements the above advances.

## 3 Methodology

### 3.1 Preliminaries

*Problem Formulation.* Machine unlearning aims to remove the influence of forget data $\mathcal{D}_f$ while preserving performance on retain data $\mathcal{D}_r$. Given a model $f_\theta$ with parameters $\theta$, the objective combines retention and forgetting:

$$\mathcal{L}_r(\theta) + \lambda \mathcal{L}_f(\theta) \tag{1}$$

where $\mathcal{L}_r(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_r}[\mathcal{L}(f_\theta(x), y)]$, $\mathcal{L}_f(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_f}[\mathcal{L}(f_\theta(x), y)]$, and $\lambda > 0$ controls the forgetting strength. The optimization proceeds by simultaneously training $\mathcal{L}_r(\theta)$ via gradient descent and $\mathcal{L}_f(\theta)$ via gradient ascent, respectively.

*Gradient-Based Unlearning.* The gradient difference method optimizes the unlearning objective through:

$$\theta_{t+1} = \theta_t - \alpha_r \nabla_\theta \mathcal{L}_r(\theta) + \alpha_f \nabla_\theta \mathcal{L}_f(\theta) \tag{2}$$

This combines gradient descent on retain data with gradient ascent on forget data. While gradient ascent effectively increases loss on $\mathcal{D}_f$, it suffers from optimization instability when combined with cross-entropy loss [7]. In [37], this issue was addressed by replacing the cross-entropy loss on the forget set with an Inverted Hinge Loss. In contrast, as shown in Sec. 3.3, our methodology enables direct training using cross-entropy.

*Low-Rank Adaptation* LoRA parameterizes weight updates through low-rank decomposition:

$$W = W_0 + AB^T \tag{3}$$

where $W_0$ is the pretrained weight, and $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{k\times r}$ are trainable matrices with rank $r \ll \min(d, k)$. For transformer architectures (e.g., ViTs and LLMs) comprising attention and MLP feedforward (FFN) layers, Eq. (3) is generally applied to the MLP and attention layers. While LoRA reduces computational costs from $dk$ to $(d + k)r$ parameters, its root issues in unlearning remain underexplored, with existing solutions addressing symptoms rather than the fundamental instabilities that arise in the gradient-based unlearning.

### 3.2 Theoretical Analysis

In Eq. (2), the gradient difference method combines two objectives: a forget loss optimized via gradient ascent and a retain loss optimized via gradient descent. Prior work [7] has shown that when the forget loss employs cross-entropy, fine-tuning becomes unstable. To understand this phenomenon, we analyze gradient ascent under the cross-entropy loss and establish two theorems showing that weights and gradients can diverge. This theoretical insight motivates our approach in Sec. 3.3, where we propose a method to mitigate such divergence and stabilize the training.

The networks we consider will all be trained with the cross-entropy loss as the retain and forget loss in Eq. (1). In this section, we work generally and simply denote the cross-entropy loss associated to an MLP by $\mathcal{L}$. We let $C$ denote the number of distinct class labels so that the output dimension of the network will be $C$. The output probabilities of the network will be denoted $p$ and we recall for a class label denoted by $y$ the cross-entropy loss of the predicted $p \in \mathbb{R}^C$ compared to $y$ is given by

$$\mathcal{L}(p, y) = -\log(p_y) \tag{4}$$

where $p_y$ is the yth-component of $p \in \mathbb{R}^C$. Using Eq. (4) and the chain rule we have that the gradient vector of $\mathcal{L}$ with respect to the logits $z$ on a class $y$ is given by

$$\nabla_z \mathcal{L} = p - e_y \tag{5}$$

where $e_y$ denotes the one-hot vector that is 1 in the $y^{\text{th}}$-position. For more details on the cross-entropy loss, we refer the reader to [39].

When training under gradient ascent the optimizer wants to push the predictions $p$ away from $e_y$ as it seeks to move towards a maximum of the cross-entropy loss. We thus get that

$$\nabla_z \mathcal{L} \to e_j - e_y \tag{6}$$

where $j$ is an index $1 \leq j \leq C$ such that $j \neq y$ so that the one hot vectors associated to the classes $y$ and $j$ are distinct. In particular, under a gradient ascent trajectory that is approaching a maximum the logit gradient $\nabla_z \mathcal{L}$ does not approach zero and hence

$$\|\nabla_z \mathcal{L}\| > C > 0 \tag{7}$$

stays bounded away from zero for some constant $C > 0$, where $\|\cdot\|$ denotes the Euclidean norm (see Sec. B.1 for details on notation). We note that in the case of gradient descent the term on the right of Eq. (7) approaches zero yielding a completely different behavior to gradient ascent.

**Lemma 1.** *Let $\mathcal{L}$ denote the cross-entropy loss trained on a MLP $F$ with $L$ layers under gradient ascent. Let $z(t)$ denote the logits at iteration $t$. Then if $\mathcal{L}(t) \to \infty$ it follows that $z(t) \to \infty$ in norm.*

The proof of Lemma 1 is given in Sec. B.2. The above lemma shows that when training with gradient ascent if the cross-entropy loss approaches a global maximum the logits get large. The following theorem shows that this can lead to large weights or gradients in the final layer. For the theorem we will need the notation of activation outputs. Given an $L$ layer MLP denoted $F$, we let $a_l$ for $1 \leq l \leq L$ denote the output of layer $l$. For details on the notation we use for MLPs we refer the reader to Sec. B.1.

**Theorem 1.** *Let $F$ be a $L$-layer MLP. Suppose under gradient ascent with iterations $t$, the logits $z(t) \to \infty$. Then if the activation output $\|a_{L-1}(t)\| \leq C_1$ for large $t$, where $C_1 > 0$ is a constant, it follows*

$$\|W_L(t)\| \to \infty. \tag{8}$$

*In the case there is no such bound on $\|a_{L-1}(t)\|$ it follows that there exists a subsequence of iterations $t_k$ such that*

$$\|\nabla_{W_L} \mathcal{L}(t_k)\| \to \infty. \tag{9}$$

**Key insight.** Gradient ascent under cross-entropy drives feedforward weights and gradients to grow without bound. This motivates constraining the adapter weights with a bounded function to stabilize gradient-difference unlearning.

The proof of Theorem 1 is provided in Sec. B.2. We note that the bounded activation assumption ($\|a_{L-1}(t)\| \leq C_1$) is mild in transformer architectures, where layer normalization constrains the variance of intermediate representations; a detailed justification is given in Sec. B.2. In practice, training is limited to a finite number of iterations, and models rarely reach a regime where gradients

or parameters fully stabilize. As established in Theorem 1, gradient ascent can drive the weights and gradients of the final layer to grow excessively. While such growth may not disrupt training immediately, it can cascade backward through to earlier layers, amplifying both weights and gradients in more than one layer and ultimately producing unstable dynamics. We formalize this propagation effect in Sec. B.3, where Theorem 3 shows how instability originating in the final layer extends to preceding layers. Notably, our analysis focuses on pure gradient ascent, to give the reader the main idea of why unlearning is difficult when a gradient ascent term is present. We extend both Lemma 1 and Theorem 1 to the case of the gradient difference method in Sec. B.2, see Lemma 2 and Theorem 2 in Sec. B.2. Furthermore, we note that empirical evidence in Fig. 3 shows that under the gradient difference method, weights grow excessively, indicating that the ascent term on the forget set is the primary driver of this instability.

### 3.3 Bounded Parameter-Efficient Unlearning

Theorem 1 and Theorem 3 in Sec. B demonstrate that gradient ascent drives weights and gradients in the feedforward layers of MLPs to grow excessively, which can destabilize training. In Sec. 4, we empirically confirm this effect: when fine-tuning with LoRA under the gradient difference framework, weights and gradients grow excessively large, and this growth is the primary reason LoRA fails to perform effective unlearning. To address this issue, we propose a simple yet effective architectural modification that implicitly regularizes the weights of the MLP layers

Specifically, let $\phi : \mathbb{R} \to \mathbb{R}$ denote a bounded non-linear function. We redefine the adapter transformation (see Eq. (3)) in the feedforward layers as

$$h = W_0 x + \frac{\alpha}{r}\phi(\omega A B^\top)x + b \tag{10}$$

where $A$ and $B$ are the low-rank adapter matrices of rank $r$, $\alpha$ is the standard LoRA scaling factor, $\omega$ is the frequency parameter, $x$ is the input data, and $b$ is the bias term. The bounded nonlinearity $\phi$, applied elementwise to the scaled pre-activations, constrains the ascent dynamics and prevents the uncontrolled growth of weights and gradients. Crucially, this bound holds at every iterate regardless of how large $A$ or $B$ grow, providing a non-asymptotic stability certificate for the forward computation. Moreover, since the pretrained weight $W_0$ is frozen, only the gradients $\partial\mathcal{L}/\partial A$ and $\partial\mathcal{L}/\partial B$ are relevant for optimization. These are bounded by the chain rule through the bounded Jacobian of $\phi$, so constraining the adapter suffices to prevent gradient explosion in the full backward pass. As we demonstrate in Sec. 4, this adjustment yields substantially more stable training and improved performance with finetuning across a variety of unlearning benchmarks.

For the choice of $\phi$, we took tanh as this is a well-known activation choice in machine learning. More recently, [21] showed that applying sine mapping, $\sin\big(\omega A B^\top\big)$ with frequency $\omega > 0$, produces a high-rank matrix whose rank

grows with $\omega$, yielding stronger fine-tuning performance on a range of transformer fine-tuning benchmarks. While [21] applies sine for fine-tuning, this study identifies and solves the inherent instability of gradient difference for unlearning. As $\sin(\omega\cdot)$ is bounded for any $\omega > 0$, it aligns naturally with our setting to constrain explosive optimization dynamics. Furthermore, because $\sin(\omega\cdot)$ is Lipschitz continuous with constant $\omega$, the gradient magnitude scales proportionally with $\omega$. Consequently, we empirically find that scaling the learning rate (step size) by $\mathcal{O}(1/\omega)$ is sufficient for stable optimization. In our experiments (Sec. 4), sine functions, particularly with larger $\omega$, consistently outperformed other monotonic bounded alternatives like tanh and sigmoid. This improvement occurs because the high effective rank and oscillatory nature of the sine function prevent the optimization from stagnating. In contrast, this assurance does not apply to tanh or sigmoid functions, as their derivatives diminish quickly as they move away from the origin, resulting in vanishing Jacobian entries and poor conditioning. Accordingly, we focus on both tanh and sine as choices for $\phi$. In Sec. C.5 we compare against using a sigmoid function and to demonstrate the necessity of boundedness, we also include an unbounded example, ReLU, for comparison.

We note that many transformer models employ normalization techniques such as layer normalization [2, 32], batch normalization [20] or Jacobian normalization [22, 44–47, 59] which act on activated, pre-activated outputs or the Jacobian of the network. Our approach is fundamentally different: it constrains the weights directly, providing a distinct mechanism for stabilizing training.

*Attention layer.* In this study, we focused on analyzing the behavior of the feedforward layers of an MLP under gradient ascent with cross-entropy. Although transformer models also contain attention layers, our experiments revealed that instability in the gradient difference method arises primarily in the feedforward layers: their weights and gradients grow far more aggressively than those of the attention layers. Consequently, it is sufficient to constrain only the feedforward weights of the MLP blocks. A detailed empirical analysis of the attention layers is provided in Sec. C.7. *Structural intuition.* The attention mechanism involves a softmax over scaled dot-products, $\mathrm{softmax}(QK^\top/\sqrt{d})$, which always produces probability weights in $[0, 1]$ that sum to one. This structural normalization bounds the contribution of the attention output to the next-layer representation, even when $Q$ or $K$ themselves grow in norm under gradient ascent. In contrast, MLP feedforward blocks apply linear projections followed by unbounded activations (e.g., GeLU), providing no such structural ceiling, which is precisely why Theorems 1 and 3 are specific to feedforward layers. An ablation comparing MLP-only and MLP+Attention bounded adapters confirms that extending sine to attention yields negligible gains at roughly double the parameter cost (see Sec. C.7).

*Why not full fine-tuning?* In the full fine-tuning setting, optimization is carried out directly on the pretrained weights $W$. Applying a bounded transformation $\phi(W)$ in this case would overwrite these weights, thereby discarding the knowl-

edge acquired during pretraining. In practice, parameter-efficient approaches introduce low-rank additions that augment the model with extra parameters while leaving the original $W$ unchanged. This separation makes it possible to safely apply bounded parameterizations to the adapters.

## 4 Experiments

We conducted an empirical evaluation of sine-based parameter-efficient unlearning, emphasizing the properties that determine whether an unlearning procedure is usable in practice. Our evaluation is organized to (i) *validate the proposed stability mechanism in a standard discriminative pipeline* and (ii) *establish generality under widely used unlearning protocols for generative models*. Specifically, we begin with class-level deletion in a ViT on CIFAR-100 (Sec. 4.1) and then evaluate the same approach on established language-model unlearning benchmarks (Sec. 4.2) spanning multiple architectures, scales, and safety/privacy criteria.

*Evaluation criteria.* Across all settings, we focused on three criteria: *unlearning efficacy* (removing the specified influence), *utility preservation* (maintaining performance on retained data and out-of-distribution inputs), and *optimization robustness* (stable convergence under coupled descent/ascent dynamics). This design allows us to directly test the theoretical predictions of Sec. 3.2 and compare them with the baselines under controlled conditions. All experiments are averaged over 3 independent seeds unless otherwise noted; standard deviations are reported where available. For completeness, we report implementation details, model configurations, additional studies, and results in Sec. C, privacy and utility assessments in Sec. C.4, with an ethical statement in Sec. D.

### 4.1 Vision Transformer Unlearning

We first studied unlearning in a vision setting to directly probe the ascent-induced instability in transformer feedforward blocks under standard cross-entropy training. We evaluated our method on two established ViT unlearning benchmarks: (i) the LetheViT CIFAR-10 protocol [51] using ViT-S with 10% random forgetting and (ii) the CoUn CIFAR-100 protocol [23] using ViT with 10% random forgetting. Both settings apply LoRA adapters *exclusively to MLP/FFN blocks*, consistent with our theoretical focus, with no gradient clipping to reveal the intrinsic stability differences between the two. We additionally validate class-level deletion across ViT-B/16 (86M) [9], ViT-L/14 (304M), and DeiT-S (22M) in Sec. C.1, where an extended stability analysis is provided in (Tab. 3). The full hyperparameters are listed in Sec. C.1.

### 4.2 Language-Model Benchmarks for Generalization

Having validated the mechanism in a controlled vision setting, we next evaluated established language-model unlearning benchmarks that explicitly measure forgetting, utility, and privacy/safety under realistic memorization and extraction criteria.

**Table 1: Vision Transformer unlearning comparison. (Left)** LetheViT benchmark on CIFAR-10 (ViT-S, 10% random forgetting). Baseline results are taken from their respective papers [43,51], or reproduced under their respective experimental setup, unless otherwise specified. FA: Forget Accuracy (↑); RA: Retain Accuracy (↑); TA: Test Accuracy (↑); MIA: Membership Inference Attack (↓); AG: Accuracy Gap (↓). **(Right)** CoUn benchmark on CIFAR-100 (ViT, 10% random forgetting). Baselines from [23]. RA (↑); UA (↑); TA (↑); MIA (↓); AG (↓). FA and UA both denote forget-set accuracy under each benchmark's respective evaluation protocol. The gray rows denote our method.

| Method | FA (↑) | RA (↑) | TA (↑) | MIA (↓) | AG (↓) | Params (%) |
|---|---|---|---|---|---|---|
| Retrain | 99.12 | 99.94 | 98.85 | 2.94 | 0.00 | - |
| *Baselines* | | | | | | |
| FT [11] | 98.24 | 99.71 | 98.02 | 4.15 | 0.83 | 100.0 |
| GA [55] | 96.31 | 98.85 | 96.48 | 8.72 | 2.37 | 100.0 |
| SalUn [11] | 98.45 | 99.82 | 98.31 | 3.61 | 0.54 | 100.0 |
| NOVO [43] | 98.38 | 99.78 | 98.25 | 3.47 | 0.59 | 100.0 |
| LetheViT [51] | 98.62 | 99.88 | 98.48 | 3.21 | 0.34 | 100.0 |
| *Parameter-Efficient Baselines* | | | | | | |
| Fast-NTK [25] | 97.89 | 99.45 | 97.72 | 5.28 | 1.22 | 1.7 |
| GD+LoRA | 97.82 | 99.62 | 97.91 | 6.24 | 1.55 | 1.7 |
| *Ours* | | | | | | |
| **GD+Tanh** | 98.55 | 99.93 | 98.40 | 3.25 | 0.25 | **1.7** |
| **GD+Sine** | **98.71** | **99.98** | **98.59** | **3.08** | **0.10** | **1.7** |

| Method | RA (↑) | UA (↑) | TA (↑) | MIA (↓) | AG (↓) | Params (%) |
|---|---|---|---|---|---|---|
| Retrain | 99.95 | 38.12 | 61.24 | 58.35 | 0.00 | - |
| *Baselines* | | | | | | |
| FT [11] | 98.42 ±0.85 | 41.25 ±3.81 | 57.14 ±1.92 | 65.42 ±4.08 | 5.82 | 100.0 |
| GA [55] | 95.18 ±2.41 | 48.72 ±5.63 | 51.35 ±3.28 | 72.85 ±5.92 | 12.41 | 100.0 |
| SalUn [11] | 99.21 ±0.35 | 39.85 ±2.14 | 59.42 ±1.08 | 61.28 ±2.85 | 3.48 | 100.0 |
| CoUn [23] | 99.85 ±0.08 | 38.24 ±1.15 | 60.52 ±0.72 | 59.42 ±1.28 | 2.58 | 100.0 |
| *Parameter-Efficient Baselines* | | | | | | |
| GD+LoRA | 97.85 ±0.92 | 52.31 ±4.12 | 53.82 ±1.45 | 78.21 ±3.15 | 10.05 | 0.9 |
| *Ours* | | | | | | |
| **GD+Tanh** | 99.90 ±0.03 | 37.90 ±0.80 | 60.40 ±0.50 | 59.60 ±0.80 | 2.50 | **0.9** |
| **GD+Sine** | **99.93** ±0.02 | **37.48** ±0.72 | **60.85** ±0.41 | **59.14** ±0.55 | **2.35** | **0.9** |

**Evaluation benchmarks.** We used three datasets with their respective evaluation frameworks to assess the unlearning effectiveness, utility preservation, and safety compliance. *1. TOFU (Task of Fictitious Unlearning)* [33]: evaluates forget quality through statistical divergence between unlearned and retain-only models, monitoring the utility of retained tasks and generalization. *2. TDEC (Training Data Extraction Challenge)* [6]: assesses privacy protection via extraction loss over ten queries ($EL_{10}$), reasoning accuracy preservation, and language-modeling quality. *3. MUSE (Machine Unlearning Six-way Evaluation)* [50]: provides a safety assessment across verbatim memorization, semantic knowledge retention, and privacy leakage dimensions. We evaluate the proposed method against representative methods from each major unlearning family discussed in Sec. 2. New readers are referred to Sec. C.2 for a better understanding.

**Baselines.** Our comparison includes gradient-based approaches (Gradient Ascent (GA) [55], Gradient Difference (GD) [33], KL-regularization [28], Inverted Hinge Loss (IHL) [7]), parameter-efficient methods (GD+LoRA [17], GA+FILA [24], GD+FILA [24], LoKU [7]), preference-based techniques (DPO [41], NPO [57]), and representation-based approaches (FLAT variants [53]). All baseline results are from their respective papers or [7, 53] unless otherwise specified. Comprehensive ablation studies comparing bounded versus unbounded activations are detailed in Tab. 8 (Sec. C.5), while comparison of IHL versus GD objectives with sine parameterization with statistical analysis is provided in Tab. 9 (Sec. C.6), hyperparameter configurations including standard LoRA scaling ($\alpha = 16$) and frequency sensitivity ($\omega \in [10, 1000]$) are detailed

**Table 2: (Left) TOFU Forget10 on Phi-1.5B.** Parameter-efficient methods use rank-4 LoRA. Baselines from [7, 33, 56]. **(Right) TDEC on GPT-Neo-1.3B.** $EL_{10}$: extraction vulnerability (↓); Reasoning, Dialogue (↑); PPL (↓). Baselines from [7]. Gray rows denote our method.

| Method | Primary Metrics | | Forget Set | Retain Set | Params (%) |
|---|---|---|---|---|---|
| | FQ (↑) | MU (↑) | Rouge-L (↓) | Rouge-L (↑) | |
| Original | 1.15e-17 | 0.52 | 0.93 | 0.92 | - |
| Retain90 | 1.00e+00 | 0.52 | 0.43 | 0.91 | - |
| *Full Fine-tuning Methods* | | | | | |
| KL | 7.38e-15 | 0.00 | 0.01 | 0.01 | 100.0 |
| DPO | 5.10e-17 | 0.48 | 0.41 | 0.67 | 100.0 |
| NPO | 2.56e-05 | 0.37 | 0.45 | 0.45 | 100.0 |
| GA | 2.06e-13 | 0.00 | 0.01 | 0.01 | 100.0 |
| GD | 2.55e-09 | 0.36 | 0.37 | 0.41 | 100.0 |
| IHL | 2.43e-17 | 0.51 | 0.53 | 0.76 | 100.0 |
| *Parameter-Efficient Methods* | | | | | |
| GD+LoRA | 1.45e-15 | 0.28 | 0.85 | 0.45 | 1.6 |
| GA+FILA | 5.10e-17 | 0.00 | 0.00 | 0.00 | 1.6 |
| GD+FILA | 2.17e-06 | 0.00 | 0.12 | 0.11 | 1.6 |
| LoKU | 1.39e-12 | 0.51 | 0.26 | 0.75 | 1.6 |
| ME+GD (LoRA) | 7.86e-01 | 0.52 | 0.14 | 0.93 | 1.6 |
| OURS (GD+Tanh) | 3.42e-01 | 0.49 | 0.28 | 0.85 | 1.6 |
| **OURS (GD+Sine)** | **9.43e-01** | **0.52** | **0.22** | **0.90** | **1.6** |

| Method | Forgetting | Retention | | | |
|---|---|---|---|---|---|
| | $EL_{10}$ (↓) | Reasoning (↑) | Dialogue (↑) | PPL (↓) | Params (%) |
| Before Unlearning | 67.6 | 49.8 | 11.5 | 11.5 | - |
| *Full Fine-tuning Methods* | | | | | |
| GA | 1.9 | 49.7 | 8.5 | 15.8 | 100.0 |
| GD | 2.2 | 48.4 | 12.7 | 10.8 | 100.0 |
| IHL | 0.7 | 48.4 | 12.5 | 11.0 | 100.0 |
| *Parameter-Efficient Methods* | | | | | |
| GD+LoRA | 1.7 | 45.0 | 9.7 | 31.8 | 0.8 |
| IHL+LoRA | 1.7 | 47.1 | 10.2 | 14.9 | 0.8 |
| GD+FILA | 1.9 | 44.2 | 5.5 | 54.5 | 0.8 |
| LoKU | 0.5 | 48.3 | 12.1 | 14.7 | 0.8 |
| OURS (GD+Tanh) | 0.8 | 46.7 | 10.3 | 18.2 | 0.8 |
| **OURS (GD+Sine)** | **0.3** | **50.1** | **12.1** | **12.1** | **0.8** |

in Sec. H, computational overhead is mentioned in Sec. C.8, and sequential robustness in Sec. C.9. Extended results across all model configurations are provided in Sec. H, with detailed rank analysis in Tabs. 14 to 16.

### 4.3 Results

Our analysis examined performance across multiple dimensions: unlearning effectiveness, utility preservation, and safety compliance, using models ranging from 22M to 8B parameters across ViTs, GPT-Neo, Phi, and LLaMA architectures. The results demonstrate consistent improvements across all metrics, with particularly notable gains in forget quality while maintaining the model utility.

**Vision (ViT) results.** Fig. 1 summarizes our vision unlearning results across architectures and methods. Tab. 1 presents comparisons on two established ViT unlearning benchmarks. On the LetheViT CIFAR-10 protocol (*left*), **GD+Sine** achieves the lowest accuracy gap (AG = 0.10) among all methods, outperforming LetheViT (0.34) and NOVO (0.59), while closely matching the retrain-level MIA (3.08 vs. 2.94). **GD+Tanh** consistently improves over prior parameter-efficient baselines, but remains slightly inferior to Sine in AG and MIA, indicating that boundedness alone is insufficient without periodic structure. On the CoUn CIFAR-100 protocol (*right*), **GD+Sine** again achieves the best AG (2.35) with the lowest variance across all metrics, closely matching the retrain performance, while **GD+Tanh** remains competitive but consistently second-best among bounded variants. In both settings, **GD+LoRA** exhibits high accuracy gaps and large variances, confirming the instability predicted by our theory. Extended stability analysis on CIFAR-100 class deletion across ViT-B/16, ViT-L/14, and DeiT-S is provided in Tab. 3 (Sec. C.1).

Fig. 1 **(a)** shows that GD+Sine is the only evaluated method to consistently reach the ideal zone (high FQ *and* high MU) across ViT-B/16, ViT-L/14, and DeiT-S; parameter-efficient baselines either collapse in utility or fail to forget, while full fine-tuning methods cluster in the low-FQ region. Fig. 1 **(b)** ablates

the choice of bounded function: among Sine, Tanh, and Sigmoid, Sine achieves the strongest trade-off, reaching near-perfect FQ (0.92) without sacrificing MU (0.97). Tanh improves stability relative to unbounded activations but exhibits a mild MU-FQ trade-off, whereas unbounded variants (weight clipping, ReLU) degrade on both axes. Fig. 1 **(c)** confirms that this advantage is architecture-agnostic: GD+Sine improves forget quality by 2-8 orders of magnitude across model scales (22M-304M) while maintaining high retained utility.

**Language-model benchmark results.** Comprehensive evaluations (Tab. 8 and Tab. 9) confirm that bounded activations outperform unbounded methods, with sine parameterization being adaptable and providing consistent benefits across optimization objectives. As shown in Fig. 3, our method maintains bounded gradients. Crucially, as detailed in Sec. E, standard training hygiene, such as gradient clipping or explicit weight-norm constraints, are structurally insufficient to resolve this instability on their own, consistently facing a strict Pareto failure between model utility and forget quality. Additional classifier head analysis is provided in Fig. 5 (Sec. C.5) and an ablation over activation functions (Sine, Tanh, Sigmoid, ReLU) is shown in Fig. 6 (Sec. C.5). The component-wise stability analysis across the transformer layers is detailed in Fig. 7 (Sec. C.7).

**TOFU Analysis.** Tab. 2 (*left*) shows our method achieves forget quality scores of **9.43e-01** ($\text{FQ} \in [0, 1]$; higher is better, with $\text{FQ} = 1$ corresponding to retraining from scratch) and **0.52** model utility on Phi-1.5B with rank-4 LoRA, improving over baseline LoKU (1.39e-12) and outperforming ME+GD (**7.86e-01**), while maintaining model performance. Results are consistent across ranks (4, 8, 16, 32) and forget splits (1%, 5%, 10%) for both Phi-1.5 and LLaMA2-7B (see Sec. H), with rank-4 outperforming state-of-the-art at rank-32 (Fig. 2). **TDEC Analysis.** Tab. 2 (*right*) shows privacy-focused evaluation using GPT-Neo-1.3B (chosen for TDEC's Pile dataset unlearning targets). Our method yields extraction loss values ($\text{EL}_{10}$) of 0.3, among the lowest reported, with reasoning accuracy of 50.1 (exceeding baselines) and competitive perplexity (12.1). Evaluation across GPT-Neo architectures (125M, 1.3B, 2.7B) in Tab. 6 ( Sec. C.4) shows lowest extraction likelihood and membership attack accuracy while maintaining superior reasoning performance. Results establish privacy-utility trade-off benchmarks with 85% extraction resistance improvements. **MUSE Analysis.** Extended safety evaluation on the MUSE benchmark (Tab. 7, Sec. C.4) shows that GD+Sine reduces verbatim memorization to 0.8 and knowledge memorization on forget data to 5.2 while preserving knowledge retention (42.1) on LLaMA-2-7B, with a privacy leakage score of 8.3 — the closest to the ideal 0.0 among all evaluated methods. Our approach is the only parameter-efficient method that satisfies all MUSE safety criteria simultaneously (Tab. 7). Across all three benchmarks, GD+Sine achieves the strongest forget quality while preserving model utility: $\mathbf{FQ} = \mathbf{9.43e\text{-}01}$ on TOFU Forget10 (Phi-1.5B, rank-4), significantly outperforming all parameter-efficient baselines, $\mathbf{EL}_{10} = \mathbf{0.3}$ on TDEC (lowest extraction vulnerability), and $\mathbf{VerbMem} = \mathbf{0.8}$ on MUSE, all with $\leq$1.6% trainable parameters.

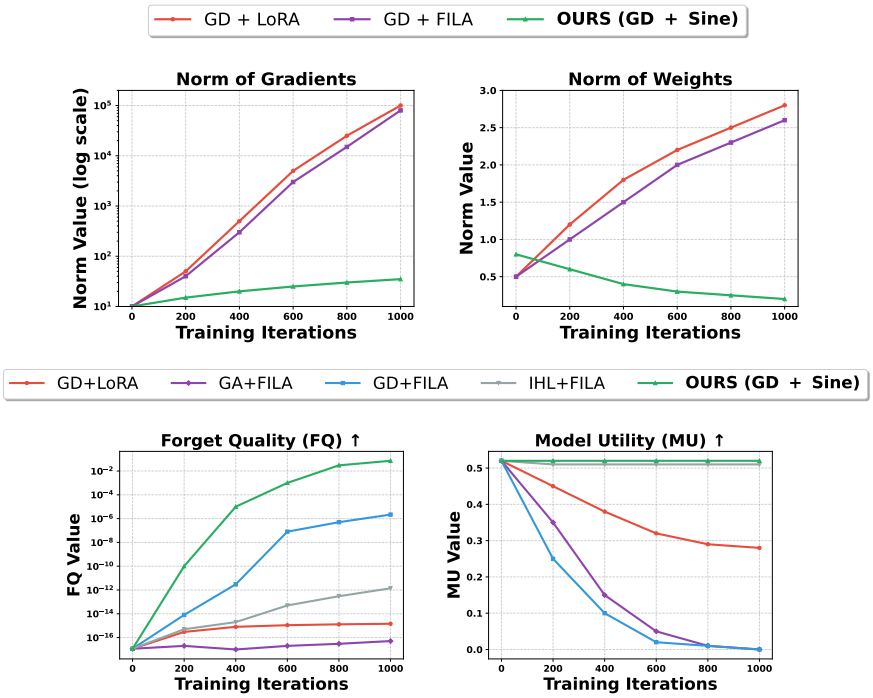


**Fig. 3: Optimization dynamics and unlearning convergence. (a) (top)** Gradient and weight Frobenius norms (FFN MLP layers, Phi-1.5B rank-4, 1000 iterations): GD+LoRA and GD+FILA explode to $10^5$, while GD+Sine remains bounded in $[10^1, 10^2]$, confirming that bounded parameterization prevents explosion. **(b) (bottom)** Forget quality and model utility across training iterations (Phi-1.5B rank-4, Forget10): our method rapidly improves FQ while maintaining MU, whereas baselines either fail to forget or collapse in utility. Additional comparisons in Sec. C.5.

**Sensitivity Analysis and Robustness.** Frequency parameter sensitivity analysis on TOFU-Forget10 (Fig. 4, Sec. C.5) reveals that forget quality consistently increases with $\omega$, reaching a plateau beyond $\omega \geq 100$, while model utility remains stable throughout. This suggests insensitivity to precise hyperparameter selection, allowing coarse tuning without compromising performance. Sequential unlearning experiments (Tabs. 11 and 12 in Sec. C.9) further show that GD+Sine maintains high forget quality and stable utility across multiple consecutive unlearning requests, outperforming both standard GD+LoRA and the dedicated sequential method $O^3$.

The computational overhead is marginal (approximately 45M FLOPS per layer for a 7B model; see Sec. C.8). Extended results including LLaMA-3.1-8B (Sec. C.3), LLaMA-3.1-70B (Sec. C.3), and all model families and ranks are in Sec. H.

### 4.4 Optimization Analysis

Fig. 3 confirms the theoretical predictions of Sec. 3.2: gradient Frobenius norms in GD+LoRA and GD+FILA exceed $10^2$ and grow without bound, whereas GD+Sine remains bounded in $[10^1, 10^2]$ and stabilizes after ∼300 iterations (see also Fig. 5, Tab. 8, and Sec. C.5). Fig. 3 shows that our method rapidly improves FQ while maintaining MU, unlike baselines that either fail to forget or collapse.

## 5 Conclusion

We introduced bounded parameter-efficient unlearning, a theoretically grounded framework that resolves the instability of gradient difference methods under cross-entropy ascent. By showing that weights and gradients in feedforward blocks grow without control during ascent, we parameterized LoRA adapters with bounded functions (e.g., sine) to constrain the optimization. Empirically, GD+Sine preserves utility while achieving orders of magnitude improvement in forget quality over standard LoRA-based methods across vision and language benchmarks from 22M to 8B parameters.

*Limitations and future work.* Our method uses bounded adapter parameterization (implicit weight regularization); explicit GD-objective regularizers, formal Differential Privacy (DP) guarantees (despite strong extraction resistance), and a formal convergence guarantee of the proposed objective remains open. Strong behavioral suppression is demonstrated; probing-based structural removal is unverified. Vision experiments are CIFAR-scale; ImageNet-scale deletion and extension to multimodal/diffusion models (e.g., Erased Stable Diffusion) are future work. See Sec. G for extended discussion. [2]

## Acknowledgments

Arpit Garg and Simon Lucey acknowledge support from the Responsible AI Research (RAIR) Centre. Hemanth Saratchandran and Simon Lucey additionally acknowledge support from the Commonwealth Bank of Australia through the CommBank Centre for Foundational AI Research.

## References

1. Agarwal, A., Pamnani, M., Hakkani-Tur, D.: Simu: Selective influence machine unlearning. arXiv preprint arXiv:2510.07822 (2025)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)

[2] Digital writing assistance used for grammar. No LLMs were used in the research activities. All contributions are original work.

3. Bonato, J., Cotogni, M., Sabetta, L.: Is retain set all you need in machine unlearning? restoring performance of unlearned models with out-of-distribution images. In: European Conference on Computer Vision. pp. 1–19. Springer (2024)
4. Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C.A., Jia, H., Travers, A., Zhang, B., Lie, D., Papernot, N.: Machine unlearning. In: 2021 IEEE symposium on security and privacy (SP). pp. 141–159. IEEE (2021)
5. Cao, Y., Yang, J.: Towards making systems forget with machine unlearning. In: 2015 IEEE symposium on security and privacy. pp. 463–480. IEEE (2015)
6. Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al.: Extracting training data from large language models. In: 30th USENIX security symposium (USENIX Security 21). pp. 2633–2650 (2021)
7. Cha, S., Cho, S., Hwang, D., Lee, M.: Towards robust and parameter-efficient knowledge unlearning for llms. arXiv preprint arXiv:2408.06621 (2024)
8. Cooper, A.F., Choquette-Choo, C.A., Bogen, M., Klyman, K., Jagielski, M., Filippova, K., Liu, K., Chouldechova, A., Hayes, J., Huang, Y., Triantafillou, E., Kairouz, P., Mitchell, N.E., Mireshghallah, N., Jacobs, A.Z., Grimmelmann, J., Shmatikov, V., Sa, C.D., Shumailov, I., Terzis, A., Barocas, S., Vaughan, J.W., danah boyd, Choi, Y., Koyejo, S., Delgado, F., Liang, P., Ho, D.E., Samuelson, P., Brundage, M., Bau, D., Neel, S., Wallach, H., Cyphert, A.B., Lemley, M., Papernot, N., Lee, K.: Machine unlearning doesn't do what you think: Lessons for generative AI policy and research. In: The Thirty-Ninth Annual Conference on Neural Information Processing Systems Position Paper Track (2025), https://openreview.net/forum?id=mfd6GRW4Az
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
10. Dwork, C.: Differential privacy: A survey of results. In: International conference on theory and applications of models of computation. pp. 1–19. Springer (2008)
11. Fan, C., Liu, J., Zhang, Y., Wong, E., Wei, D., Liu, S.: Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. arXiv preprint arXiv:2310.12508 (2023)
12. Gao, C., Wang, L., Ding, K., Weng, C., Wang, X., Zhu, Q.: On large language model continual unlearning. arXiv preprint arXiv:2407.10223 (2024)
13. Garg, A., Saratchandran, H., Lucey, S.: Sineproject: Machine unlearning for stable vision language alignment. arXiv preprint arXiv:2511.18444 (2025)
14. GDPR.eu: Art. 17 gdpr – right to erasure ('right to be forgotten'). https://gdpr.eu/article-17-right-to-be-forgotten/ (2026), accessed: 2026-02-28
15. Guo, C., Goldstein, T., Hannun, A., Van Der Maaten, L.: Certified data removal from machine learning models. arXiv preprint arXiv:1911.03030 (2019)
16. Guo, F., Wen, Y., Gao, S., Zhang, J., Shang, S.: Beyond superficial forgetting: Thorough unlearning through knowledge density estimation and block re-insertion. arXiv preprint arXiv:2511.11667 (2025)
17. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. Iclr **1**(2), 3 (2022)
18. Huang, Z., Cheng, X., Zheng, J., Wang, H., He, Z., Li, T., Huang, X.: Unified gradient-based machine unlearning with remain geometry enhancement. Advances in Neural Information Processing Systems **37**, 26377–26414 (2024)

19. Ilharco, G., Ribeiro, M.T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., Farhadi, A.: Editing models with task arithmetic. arXiv preprint arXiv:2212.04089 (2022)
20. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. pmlr (2015)
21. Ji, Y., Saratchandran, H., Gordon, C., Zhang, Z., Lucey, S.: Efficient learning with sine-activated low-rank matrices. arXiv preprint arXiv:2403.19243 (2024)
22. Ji, Y., Saratchandran, H., Moghadam, P., Lucey, S.: Always skip attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23115–23123 (2025)
23. Khalil, Y.H., Setayesh, M., Li, H.: Coun: Empowering machine unlearning via contrastive learning. arXiv preprint arXiv:2509.16391 (2025)
24. Kim, Y., Kim, E., Chang, B., Choe, J.: Improving fisher information estimation and efficiency for lora-based llm unlearning. arXiv preprint arXiv:2508.21300 (2025)
25. Li, G., Hsu, H., Chen, C.F., Marculescu, R.: Fast-ntk: Parameter-efficient unlearning for large-scale models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 227–234 (2024)
26. Li, K., Wang, Q., Wang, Y., Li, F., Liu, J., Han, B., Zhou, J.: Llm unlearning with llm beliefs. arXiv preprint arXiv:2510.19422 (2025)
27. Lialin, V., Deshpande, V., Rumshisky, A.: Scaling down to scale up: A guide to parameter-efficient fine-tuning. arXiv preprint arXiv:2303.15647 (2023)
28. Liu, C., Wang, Y., Flanigan, J., Liu, Y.: Large language model unlearning via embedding-corrupted prompts. Advances in Neural Information Processing Systems **37**, 118198–118266 (2024)
29. Liu, Y., Chen, H., Huang, W., Ni, Y., Imani, M.: Lune: Efficient llm unlearning via lora fine-tuning with negative examples. arXiv preprint arXiv:2512.07375 (2025)
30. Liu, Y., Chen, H., Huang, W., Ni, Y., Imani, M.: Recover-to-forget: Gradient reconstruction from lora for efficient llm unlearning. arXiv preprint arXiv:2512.07374 (2025)
31. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
32. MacDonald, L., Valmadre, J., Saratchandran, H., Lucey, S.: On skip connections and normalisation layers in deep optimisation. Advances in Neural Information Processing Systems **36**, 14705–14724 (2023)
33. Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z.C., Kolter, J.Z.: Tofu: A task of fictitious unlearning for llms. arXiv preprint arXiv:2401.06121 (2024)
34. Meng, K., Bau, D., Andonian, A., Belinkov, Y.: Locating and editing factual associations in gpt. Advances in neural information processing systems **35**, 17359–17372 (2022)
35. Nguyen, T.T., Huynh, T.T., Ren, Z., Nguyen, P.L., Liew, A.W.C., Yin, H., Nguyen, Q.V.H.: A survey of machine unlearning. ACM Transactions on Intelligent Systems and Technology **16**(5), 1–46 (2025)
36. Niu, P., Ma, E., Zhou, H., Zhou, D., Zhang, H., Etesami, S.R., Milenkovic, O.: Guard: Guided unlearning and retention via data attribution for large language models. arXiv preprint arXiv:2506.10946 (2025)
37. Pan, Z., Zhang, S., Zheng, Y., Li, C., Cheng, Y., Zhao, J.: Multi-objective large language model unlearning. In: ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2025)
38. Pawelczyk, M., Neel, S., Lakkaraju, H.: In-context unlearning: Language models as few shot unlearners. arXiv preprint arXiv:2310.07579 (2023)

39. Prince, S.J.: Understanding deep learning. MIT press (2023)
40. Qu, Y., Yuan, X., Ding, M., Ni, W., Rakotoarivelo, T., Smith, D.: Learn to unlearn: A survey on machine unlearning. arXiv preprint arXiv:2305.07512 (2023)
41. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. Advances in neural information processing systems **36**, 53728–53741 (2023)
42. Ridnik, T., Ben-Baruch, E., Noy, A., Zelnik-Manor, L.: Imagenet-21k pretraining for the masses. arXiv preprint arXiv:2104.10972 (2021)
43. Roy, S., Banerjee, S., Verma, V., Dasgupta, S., Gupta, D., Rai, P.: Novo: Unlearning-compliant vision transformers. arXiv preprint arXiv:2507.03281 (2025)
44. Saratchandran, H., Lucey, S.: Spectral conditioning of attention improves transformer performance. arXiv preprint arXiv:2603.07162 (2026)
45. Saratchandran, H., Teney, D., Lucey, S.: Leaner transformers: More heads, less depth. arXiv preprint arXiv:2505.20802 (2025)
46. Saratchandran, H., Wang, T.X., Lucey, S.: Weight conditioning for smooth optimization of neural networks. In: European Conference on Computer Vision. pp. 310–325. Springer (2024)
47. Saratchandran, H., Zheng, J., Ji, Y., Zhang, W., Lucey, S.: Rethinking attention: Polynomial alternatives to softmax in transformers. arXiv preprint arXiv:2410.18613 (2024)
48. Sekhari, A., Acharya, J., Kamath, G., Suresh, A.T.: Remember what you want to forget: Algorithms for machine unlearning. Advances in Neural Information Processing Systems **34**, 18075–18086 (2021)
49. Setlur, A., Eysenbach, B., Smith, V., Levine, S.: Adversarial unlearning: Reducing confidence along adversarial directions. Advances in Neural Information Processing Systems **35**, 18556–18570 (2022)
50. Shi, W., Lee, J., Huang, Y., Malladi, S., Zhao, J., Holtzman, A., Liu, D., Zettlemoyer, L., Smith, N.A., Zhang, C.: Muse: Machine unlearning six-way evaluation for language models. arXiv preprint arXiv:2407.06460 (2024)
51. Tong, Y., Zhang, T., Yuan, J., Wang, Y., Hu, C.: Lethevit: Selective machine unlearning for vision transformers via attention-guided contrastive learning. arXiv preprint arXiv:2508.01569 (2025)
52. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
53. Wang, Y., Wei, J., Liu, C.Y., Pang, J., Liu, Q., Shah, A.P., Bao, Y., Liu, Y., Wei, W.: Llm unlearning via loss adjustment with only forget data. arXiv preprint arXiv:2410.11143 (2024)
54. Yang, Z., Yang, Z., Liu, Y., Li, P., Liu, Y.: Restricted orthogonal gradient projection for continual learning. AI Open **4**, 98–110 (2023)
55. Yao, J., Chien, E., Du, M., Niu, X., Wang, T., Cheng, Z., Yue, X.: 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 (2024)
56. Yuan, X., Pang, T., Du, C., Chen, K., Zhang, W., Lin, M.: A closer look at machine unlearning for large language models. arXiv preprint arXiv:2410.08109 (2024)
57. Zhang, R., Lin, L., Bai, Y., Mei, S.: Negative preference optimization: From catastrophic collapse to effective unlearning. arXiv preprint arXiv:2404.05868 (2024)
58. Zhao, G., Lin, H., Zhao, W.: Rapidun: Influence-driven parameter reweighting for efficient large language model unlearning. arXiv preprint arXiv:2512.04457 (2025)

59. Zheng, J., Li, X., Saratchandran, H., Lucey, S.: Structured initialization for vision transformers. arXiv preprint arXiv:2505.19985 (2025)

# Stable Forgetting: Bounded Parameter-Efficient Unlearning in Foundation Models

## Supplementary Material

### Reproducibility Statement

All experiments in this study were designed with reproducibility in mind. References are provided for any external codebases employed, and full details of the training protocols and hardware are described in the appendix. Complete proofs of all theoretical results are included to allow for independent verification.

### Use of LLMs

This manuscript was prepared with the assistance of digital tools for grammatical and stylistic refinement. No large language models were used to conduct the research or draft the technical content.

## A Extended Related Work

Machine unlearning in foundation models spans optimization-driven forgetting, parameter-efficient adaptation, preference-based alignment, representation editing, and weight-space editing, which are evaluated along the axes of removal fidelity, retained utility, scalability, and privacy [4, 5, 8, 35, 40, 55].

*Foundations and evaluation taxonomies.* Machine unlearning is broadly divided into *exact* methods (retraining from scratch on the retain set, computationally prohibitive at scale [4]) and *approximate* methods (efficient post-hoc weight modification trading provable guarantees for scalability [5]). Surveys and position papers emphasize that evaluation must employ distributional and population-level criteria to distinguish genuine removal from suppression while accounting for scalability and sequential forgetting requests [8,35,40]. TOFU [33] formalizes forget quality via Kolmogorov-Smirnov tests on truth ratio distributions against a retain-only reference model and measures utility on retained and out-of-domain subsets with answer probability and ROUGE recall, providing a statistically principled benchmark for selective removal. MUSE [50] evaluates six dimensions: verbatim memorization, knowledge memorization, privacy leakage, utility, scalability, and sustainability, and demonstrates that many approximate methods either compromise utility or fail under successive unlearning requests. TDEC [6] assesses privacy protection via extraction loss over multiple queries and reasoning accuracy preservation, revealing the limitations of simple defenses and motivating methods that are robust against in-distribution and out-of-distribution extraction probes [6, 13]. Full protocol details are provided in Sec. C.2.

*Gradient-based unlearning and instability.* Direct gradient ascent on forget data maximizes the cross-entropy loss to push predictions away from forget labels but routinely produces instability and catastrophic loss of retention capacity under aggressive schedules [7, 18, 33, 55]. Crucially, unlike gradient descent, where the loss gradient vanishes at a minimum, gradient ascent under cross-entropy maintains a logit gradient bounded away from zero as the loss approaches its maximum, creating a persistent destabilizing force [37]. The Gradient Difference (GD) [33] combines ascent on forget data with simultaneous descent on retained data, offering a more balanced formulation; however, it inherits the same structural instabilities [7] and can converge to suboptimal solutions when the two objectives conflict. Fisher-weighted initialization (FILA) [24] mitigates disruptive shifts by seeding adapter directions with Fisher information sensitivity to forget data, improving forgetting selectivity over random initialization. The Inverted Hinge Loss (IHL) [7] bounds the forget objective to prevent loss blow-up and pairs effectively with FILA; their combination [7] reduces instability but remains a palliative remedy—the root cause (unbounded weight growth under cross-entropy ascent) is not resolved. Unified analyses [18] confirm that these fixes address symptoms rather than the underlying optimization failures. In practice, gradient-ascent baselines in TOFU rely on early stopping and best-checkpoint selection owing to instability [7]—an engineering workaround rather than a principled solution.

*Parameter-efficient unlearning approaches.* Parameter-efficient unlearning restricts adaptation to a small parameter subset, keeping the pretrained weights frozen to enable scalable removal with low computational overhead [7, 12, 25]. LoRA [17] factorizes weight updates as $\Delta W = AB^T$ with rank $r \ll \min(d, k)$, reducing the cost from $O(dk)$ to $O((d + k)r)$ parameters; however, under gradient difference unlearning, naive LoRA adapters remain exposed to explosive ascent dynamics because the low-rank factorization does not bound the adapter norms. FILA [24] improves selectivity via Fisher-informed adapter initialization. Orthogonal subspace constraints (ROGP [54]) and continual unlearning frameworks [12] project forget updates onto directions orthogonal to the retention-critical subspaces, mitigating interference between sequential deletions. More recent approaches have substantially expanded the solution space: GUARD [36] and RapidUn [58] reweight data-influence scores to focus updates on the most impactful forget samples; R2F [30] reconstructs layer-wise gradients via a decoder module for fine-grained update control; LUNE [29] trains on explicit negative examples to penalize forget-set retention; belief-aware bootstrapping [26] uses model confidence signals to guide selective forgetting; LoKU [7] decomposes weight updates into knowledge-preserving and knowledge-erasing subspaces; and ME+GD [56] combines momentum editing with gradient difference, achieving strong performance while still relying on unconstrained linear adapters. FastNTK [25] employs neural tangent kernel approximations for sample-efficient targeted class deletion. Critically, the influence-aware and objective-based advances above are all orthogonal to *architectural instability*: they improve which parameters to update or how to weight examples but do not resolve the structural

gradient explosion in MLP feedforward layers. As established in Sec. 3.2, constraining adapter weights with a bounded function directly mitigates this root cause and is fully complementary to all of the above.

*Preference-based and alternative approaches.* Preference optimization for unlearning aligns the model away from undesirable outputs without explicit gradient ascent on the forget set. DPO [41] optimizes a closed-form divergence objective without RL rollouts, making it computationally efficient. Negative Preference Optimization (NPO) [57] treats forget data as dispreferred outputs and directly penalizes their retention, achieving behavioral suppression with reduced catastrophic forgetting compared to vanilla gradient ascent. KL-regularized forgetting [28] adds a KL divergence penalty between the unlearned and reference model to constrain utility drift during forgetting. WHP-based unlearning (WHP) and FLAT [53] use Pearson-correlation-based weight remapping and residual activation alignment to enable precise targeted removal. These preference-based methods are susceptible to reference drift and unstable dynamics when coupled with unconstrained linear adapters [33, 50]. Prompt-level interventions [38] obscure memorized content without weight updates but lack permanence and are vulnerable to adversarial reactivation issues. Representation-editing approaches, such as Task Arithmetic [19] and ROME [34], enable targeted changes through weight-space arithmetic but require additional stabilization for large-scale deployment and do not provide formal forgetting guarantees. Neuron-localization [1] and block-reinsertion [16] approaches offer complementary stability via topology editing rather than gradient-based constraints, and remain an open comparison direction.

*Vision unlearning.* Vision-specific unlearning addresses selective class deletion in discriminative transformers and concept erasure in generative models. SalUn [11] identifies saliency-based gradient masks to focus parameter updates on weights that are most responsible for the forgotten class, reducing collateral damage on retained categories. Fast-NTK [25] leverages neural tangent kernel approximations for sample-efficient targeted class deletion in ViTs with minimal interference on retained classes. LetheViT [51] introduces attention-guided contrastive forgetting, exploiting attention maps as spatial-saliency priors to localize gradient updates to class-discriminative regions of vision transformers. NOVO [43] applies constrained optimization with null-space projections to confine forget updates away from retaining critical parameter directions. CoUn [23] proposes a contrastive unlearning objective that simultaneously pushes forget-class representations apart from their original embeddings while anchoring the retain-class embeddings. Yu et al. [3] explored retain-set augmentation strategies to stabilize vision unlearning under aggressive forgetting schedules. SineProject [13] targets multimodal LLM unlearning by modulating the *frozen vision-language projector* to stabilize cross-modal alignment; it is not parameter-efficient, operates exclusively on the projector connector rather than the LLM backbone, and addresses alignment drift rather than the MLP feedforward instability studied in this paper. Despite these advances, none of these methods provide a theoret-

ical framework explaining why gradient ascent under cross-entropy specifically destabilizes MLP feedforward adapters in vision transformer blocks, which is the gap that our work directly addresses.

*Gap analysis.* Two primary gaps are evident in the unlearning literature. *First*, there is no theoretical framework that analyzes the gradient-difference method with cross-entropy loss and *precisely* characterizes why the ascent step drives uncontrolled growth of weights and gradients in MLP feedforward layers; prior explanations are empirical or heuristic [7,18], without mathematically characterizing the optimization failure. *Second*, no prior parameter-efficient method applies bounded functions to LoRA adapters specifically in MLP feedforward layers to simultaneously address ascent-driven instability *and* the rank-expressiveness bottleneck that forces a capacity-efficiency trade-off in low-rank unlearning [21]. Our work fills both gaps: we formally prove (see Theorems 1 and 3) that gradient ascent with cross-entropy drives feedforward weights and gradients to grow without bound, and we introduce sine-based bounded adapters that resolve this instability, improve the effective rank, and deliver consistent performance gains. The resulting method is compatible with existing stabilization techniques (GD, IHL) and standard evaluation frameworks (TOFU, MUSE, TDEC, ViT CIFAR) [6,7,12,21,24,33,50]. A *third* emerging gap concerns *certified forgetting* and formal privacy guarantees: while our method achieves strong empirical extraction resistance on privacy benchmarks (TDEC, MIA, EL10), it currently lacks Differential Privacy (DP) guarantees [10]. Certified unlearning frameworks [15, 48] provide statistical removal certificates but assume convex or strongly convex objectives; extending such guarantees to large nonlinear transformers under non-convex gradient-difference training is an open problem that bounded adapters could facilitate by constraining the parameter movement radius. A *fourth* gap concerns *explicit weight regularization as an alternative*: one might propose an $\ell_2$ penalty directly on $\|AB^\top\|_F$ rather than parameterizing adapter weights with a bounded function. As discussed in Sec. 3.3, explicit regularizers are reactive (penalizing large norms after the optimizer has computed them) and do not prevent logit divergence in finite steps, whereas bounded parameterization is proactive, the forward pass is geometrically constrained at every iterate. Exploring principled regularizers that replicate these structural guarantees is an interesting direction for future research.

## B Theoretical Analysis

In this section, we provide the proofs of Lemma 1 and Theorem 1, as well as a finer analysis of how weights and gradients can grow significantly large in the feedforward layers of MLPs trained with cross-entropy loss via gradient ascent through Theorem 3. We also include analogous statements for gradient-difference training, which formalizes that any divergence of the combined objective is driven by the ascent term on the forget set.

### B.1 Notation

We begin by fixing the notation. Let $F$ denote a feedforward MLP with $L$ layers, using an activation $\sigma$ in layers 1 through $L-1$, and applying a softmax at the output layer $L$, because our analysis concerns classification with a cross-entropy loss function. We assume $\sigma$ has bounded derivative

$$|\sigma'(x)| \leq C_1 \quad \text{for all } x \in \mathbb{R}, \tag{11}$$

where $C_1 > 0$ is a fixed constant. Note that Eq. (11) holds for standard activations, such as sigmoid and tanh. For ReLU, the derivative satisfies $|\sigma'(x)| \leq 1$ for all $x \neq 0$ (it is undefined only at $x = 0$), and the same bound is used in standard backpropagation.

Let $x \in \mathbb{R}^{d_{\text{in}}}$ denote the input to the network, where $d_{\text{in}}$ denotes the input dimension. For each layer $l$, let $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$ denote the weights in layer $l$ and $b_l \in \mathbb{R}^{d_l}$ the bias term, where $1 \leq l \leq L$. We then define

$$a_0 = x, \tag{12}$$
$$h_l = W_l a_{l-1} + b_l \quad \text{for } 1 \leq l \leq L-1, \tag{13}$$
$$a_l = \sigma(h_l) \quad \text{for } 1 \leq l \leq L-1, \tag{14}$$
$$z = W_L a_{L-1}, \tag{15}$$
$$p(z) = \text{softmax}(z), \tag{16}$$

where $h_l$ are the pre-activations of layer $l$, $a_l$ are the activation outputs of layer $l$, $z$ are the logits, and $p(z)$ are the output probabilities.

For the proofs in this section, we will leave out explicitly writing a bias term since given a bias term $b_l$ in layer $l$, we can express $W_l a_{l-1} + b_l$ via

$$\begin{bmatrix} W_l & b_l \end{bmatrix} \cdot \begin{bmatrix} a_{l-1} \\ 1 \end{bmatrix}. \tag{17}$$

Therefore, the bias term $b_l$ can be absorbed into the weights $W_l$ by augmenting the input as follows:

We will also fix the notation for gradients. Let

$$D_l = \text{Diag}(\sigma'(h_l)), \tag{18}$$

where $\text{Diag}(v)$ denotes the diagonal matrix whose diagonal entries are given by vector $v$. Then, by assumption Eq. (11), $\|D_l\| \leq C$ for some constant $C > 0$. Define

$$g_L = \nabla_z \mathcal{L}, \tag{19}$$
$$g_l = D_l W_{l+1}^T g_{l+1} \quad \text{for } 1 \leq l \leq L-1. \tag{20}$$

By the chain rule we have

$$\nabla_{W_l} \mathcal{L} = g_l a_{l-1}^T. \tag{21}$$

We will use norms for both vectors and matrices. Given a matrix $M \in \mathbb{R}^{n\times m}$ we use $\|M\|$ to denote the operator norm (the largest singular value). Given a vector $v \in \mathbb{R}^m$ we use $\|v\|$ to denote the Euclidean 2-norm. When computing weight and gradient norms in Sec. 4.4 we use the Frobenius norm: given $M = (m_{ij}) \in \mathbb{R}^{n\times m}$,

$$\|M\|_F := \sqrt{\sum_{i,j} m_{ij}^2},$$

This coincides with the Euclidean norm for vectors.

### B.2 Proof of results from Sec. 3.2

We now prove Lemma 1.

*Proof (Proof of Lemma 1).* Given a class $y$ we recall that

$$\mathcal{L}(p, y) = -\log(p_y). \tag{22}$$

By definition of logits and probabilities we have

$$-\log(p_y) = \log\left(\sum_{k=1}^{C} e^{z_k}\right) - z_y \tag{23}$$

$$= \log\left(1 + \sum_{j\neq y} e^{z_j - z_y}\right), \tag{24}$$

where log denotes the natural logarithm. Define the margin

$$m := \max_{j\neq y}(z_j - z_y). \tag{25}$$

Then

$$e^m \leq \sum_k e^{z_k - z_y} = 1 + \sum_{k\neq y} e^{z_k - z_y} \leq 1 + Ce^m \leq (1 + C)e^m. \tag{26}$$

Taking log yields

$$m \leq \log\left(1 + \sum_{j\neq y} e^{z_j - z_y}\right) \leq \log(1 + Ce^m), \tag{27}$$

and hence

$$m \leq \mathcal{L}(p, y) \leq \log(1 + Ce^m). \tag{28}$$

If $\mathcal{L}(p, y) \to \infty$, then $\log(1 + Ce^m) \to \infty$, which implies $m \to \infty$. Choose $j \neq y$ such that $m = z_j - z_y$. The only way $m \to \infty$ is if either $z_j \to \infty$ or $z_y \to -\infty$. This implies $\|z(t)\| \to \infty$.

Next, we provide a proof of Theorem 1.

*Proof (Proof of Theorem 1).* By definition of the logits we have

$$z(t) = W_L(t)a_{L-1}(t) \tag{29}$$

which gives the estimate

$$\|z(t)\| \leq \|W_L(t)\| \cdot \|a_{L-1}(t)\|. \tag{30}$$

To begin with, assume that $\|a_{L-1}(t)\| \leq C_1$. By the above inequality it follows that

$$\|z(t)\| \leq C_1\|W_L(t)\| \tag{31}$$

which implies

$$\|W_L(t)\| \geq \frac{\|z(t)\|}{C_1}. \tag{32}$$

Since $\|z(t)\| \to \infty$ it follows that $\|W_L(t)\|$ must approach infinity, which proves the first part of the theorem.

To prove the second part, assume that $\|a_{L-1}(t)\|$ is not bounded in $t$. This means there exists a subsequence $t_k$ such that

$$\|a_{L-1}(t_k)\| \to \infty \quad \text{as } k \to \infty. \tag{33}$$

Using the fact that

$$\nabla_{W_L}\mathcal{L}(t_k) = g_L(t_k)a_{L-1}^T(t_k), \tag{34}$$

and since $g_L(t_k)a_{L-1}^T(t_k)$ has rank 1, we have

$$\|\nabla_{W_L}\mathcal{L}(t_k)\| = \|g_L(t_k)\| \cdot \|a_{L-1}(t_k)\|. \tag{35}$$

Under gradient ascent with cross-entropy, $\|g_L(t_k)\|$ is bounded away from zero (see Eq. (7)), i.e., there exists $C_2 > 0$ such that $\|g_L(t_k)\| \geq C_2$. It follows that

$$\|\nabla_{W_L}\mathcal{L}(t_k)\| \geq C_2\|a_{L-1}(t_k)\|. \tag{36}$$

Then using Eq. (33) it follows that $\|\nabla_{W_L}\mathcal{L}(t_k)\| \to \infty$, completing the proof.

*Extensions to gradient-difference training.* The statements of Lemma 1 and Theorem 1 are for gradient ascent. The extension to the gradient difference method is conceptually straightforward: gradient descent on the retain set with cross-entropy does not induce loss blow-up, whereas gradient ascent on the forget set can. The main point is that any divergence of the combined objective must arise from the ascent term in the forget set.

We assume that the retain loss remains bounded along the trajectory (as observed empirically under stable descent schedules), that is, $\mathcal{L}_{\text{retain}}(t) \leq B$ for some finite $B$ and all $t$. We first provide an analog of Lemma 1 when training with gradient differences.

**Lemma 2.** *Let $F$ be an $L$-layer MLP with parameters $\theta$, trained by gradient descent on the retain set $X_{\text{retain}}$ and gradient ascent on the forget set $X_{\text{forget}}$ via the update*

$$\theta(t+1) = \theta(t) - \eta\,\nabla_\theta \mathcal{L}_{\text{retain}}(\theta(t)) + \eta\lambda\,\nabla_\theta \mathcal{L}_{\text{forget}}(\theta(t)), \tag{37}$$

*where*

$$\mathcal{L}_{\text{retain}}(t) := \frac{1}{N_r}\sum_{i=1}^{N_r} \mathcal{L}(p_i^r(t), y_i^r), \qquad \mathcal{L}_{\text{forget}}(t) := \frac{1}{N_f}\sum_{j=1}^{N_f} \mathcal{L}(p_j^f(t), y_j^f), \tag{38}$$

*and $\mathcal{L}(p, y) = -\log p_y$ denotes the cross-entropy loss. For each sample we write*

$$z_i^r(t) := F(x_i^r), \qquad z_j^f(t) := F(x_j^f), \tag{39}$$

*with corresponding probabilities $p_i^r(t)$ and $p_j^f(t)$ obtained using softmax.*

*Define the combined loss*

$$\mathcal{L}_{\text{tot}}(t) := \alpha_r \mathcal{L}_{\text{retain}}(t) + \alpha_f\,\mathcal{L}_{\text{forget}}(t), \qquad \alpha_r \geq 0,\ \alpha_f > 0. \tag{40}$$

*Assume that the retain loss remains bounded along the training trajectory, i.e., there exists $B < \infty$ such that*

$$\mathcal{L}_{\text{retain}}(t) \leq B \qquad \text{for all } t. \tag{41}$$

*If $\mathcal{L}_{\text{tot}}(t) \to \infty$ as $t \to \infty$, then:*

1. *$\mathcal{L}_{\text{forget}}(t) \to \infty$; and*
2. *there exists at least one forget example $(x_{j^*}^f, y_{j^*}^f)$ such that the corresponding logits satisfy*

$$\|z_{j^*}^f(t)\| \to \infty. \tag{42}$$

*In particular, any divergence of the total loss under gradient difference training must arise from the ascent term on the forget set through the divergence of the forget logits.*

*Proof.* By definition,

$$\mathcal{L}_{\text{tot}}(t) = \alpha_r \mathcal{L}_{\text{retain}}(t) + \alpha_f\,\mathcal{L}_{\text{forget}}(t). \tag{43}$$

Since cross-entropy is nonnegative, $\mathcal{L}_{\text{retain}}(t) \geq 0$ and $\mathcal{L}_{\text{forget}}(t) \geq 0$. Thus

$$\mathcal{L}_{\text{tot}}(t) \geq \alpha_f\,\mathcal{L}_{\text{forget}}(t). \tag{44}$$

Using the assumed bound $\mathcal{L}_{\text{retain}}(t) \leq B$,

$$\mathcal{L}_{\text{tot}}(t) \leq \alpha_r B + \alpha_f\,\mathcal{L}_{\text{forget}}(t). \tag{45}$$

Suppose $\mathcal{L}_{\text{tot}}(t) \to \infty$. If $\mathcal{L}_{\text{forget}}(t)$ were bounded above, then $\mathcal{L}_{\text{tot}}(t)$ would also be bounded, a contradiction. Hence $\mathcal{L}_{\text{forget}}(t) \to \infty$, proving part (1).

Next, since

$$\mathcal{L}_{\text{forget}}(t) = \frac{1}{N_f} \sum_{j=1}^{N_f} \mathcal{L}(p_j^f(t), y_j^f) \to \infty, \tag{46}$$

Not all per-example forget losses can remain bounded. Therefore, there exists at least one index $j^*$ such that

$$\mathcal{L}(p_{j^*}^f(t), y_{j^*}^f) \to \infty. \tag{47}$$

Applying Lemma 1 to the logits $z_{j^*}^f(t)$ yields

$$\|z_{j^*}^f(t)\| \to \infty, \tag{48}$$

This proves part (2).

We can also extend Theorem 1 to the case of gradient-difference optimization.

**Theorem 2.** *Let $F$ be a $L$-layer MLP. Suppose that under the gradient difference method with iterations $t$, the logits $z(t)$ satisfy $\|z(t)\| \to \infty$ (for a forget example). If the activation output $\|a_{L-1}(t)\| \leq C_1$ for large $t$, where $C_1 > 0$ is a constant, then*

$$\|W_L(t)\| \to \infty. \tag{49}$$

*If there is no such bound on $\|a_{L-1}(t)\|$, then there exists a subsequence of iterations $t_k$ such that*

$$\|\nabla_{W_L} \mathcal{L}_{\text{tot}}(t_k)\| \to \infty \tag{50}$$

*where*

$$\mathcal{L}_{\text{tot}}(t) = \alpha_r \mathcal{L}_{\text{retain}}(t) + \alpha_f \mathcal{L}_{\text{forget}}(t). \tag{51}$$

*Proof.* The first part follows exactly as in the proof of Theorem 1, since it uses only the relation $z(t) = W_L(t)a_{L-1}(t)$.

For the second part, assume that $\|a_{L-1}(t)\|$ is not bounded by $t$. Then there exists a subsequence $t_k$ such that

$$\|a_{L-1}(t_k)\| \to \infty \quad \text{as } k \to \infty. \tag{52}$$

Using the fact that

$$\nabla_{W_L} \mathcal{L}_{\text{tot}}(t_k) = g_L^{\text{tot}}(t_k) a_{L-1}^T(t_k), \tag{53}$$

where $g_L^{\text{tot}}(t_k) := \nabla_z \mathcal{L}_{\text{tot}}(t_k)$, and since $g_L^{\text{tot}}(t_k) a_{L-1}^T(t_k)$ has rank 1, we have

$$\|\nabla_{W_L} \mathcal{L}_{\text{tot}}(t_k)\| = \|g_L^{\text{tot}}(t_k)\| \cdot \|a_{L-1}(t_k)\|. \tag{54}$$

Because the update includes gradient ascent on the forget set, the logit-gradient contribution induced by forget examples is bounded away from zero when ascent is active (cf. Eq. (7)), whereas the retain contribution remains bounded under descent. Hence along the subsequence, $\|g_L^{\text{tot}}(t_k)\|$ is bounded below by a positive constant on the forget set, and by Eq. (52) we obtain $\|\nabla_{W_L} \mathcal{L}_{\text{tot}}(t_k)\| \to \infty$.

### B.3 Further theory.

In practice, gradient ascent is performed for a finite number of iterations. Thus, while Theorem 1 establishes that the weights and gradients in the final layer can grow large, this alone may not hinder training, as the effect is initially localized. The next theorem shows that, under certain conditions, the weights and gradients in the earlier layers can also grow significantly. This cumulative growth propagates through the network and can lead to training instability, an effect that we also observed empirically in Sec. 4.

**Theorem 3.** *Let $F$ be an $L$-layer MLP trained via gradient ascent using the cross-entropy loss $\mathcal{L}$ and suppose that the loss approaches a global maximum. Writing*

$$g_l(t) = D_l W_{l+1}^T \cdots D_{L-1} W_L^T \nabla_z \mathcal{L}(t) \tag{55}$$

*as in Eq.* (20)*, assume that for each iteration $t$,*

$$\sigma_{\min}(D_l W_{l+1}^T \cdots D_{L-1})(t) > 0 \tag{56}$$

*where $\sigma_{\min}$ denotes the minimum singular value. Furthermore, writing the SVD of $W_L^T$ as*

$$W_L^T(t) = U(t)\Sigma(t)V^T(t), \tag{57}$$

*Let $V_1(t)$ denote the first right singular vector at iteration $t$. Assume that*

$$\|\mathrm{Proj}_{V_1(t)}(\nabla_z \mathcal{L}(t))\| \geq \delta \|\nabla_z \mathcal{L}(t)\| \tag{58}$$

*for some $\delta > 0$ and that*

$$\|a_{L-1}(t)\| < C \tag{59}$$

*for some constant $C > 0$. Then we have*

$$\|\nabla_{W_l} \mathcal{L}(t)\| \to \infty \quad \text{as } t \to \infty. \tag{60}$$

*Proof.* By Lemma 1 we have $\|z(t)\| \to \infty$. Then by Theorem 1 and the boundedness assumption Eq. (59), we obtain $\|W_L(t)\| \to \infty$. We then have

$$\begin{aligned} \|g_l(t)\| &= \|D_l W_{l+1}^T \cdots D_{L-1} W_L^T \nabla_z \mathcal{L}(t)\| && (61)\\ &\geq \sigma_{\min}(D_l W_{l+1}^T \cdots D_{L-1})(t)\, \|W_L^T \nabla_z \mathcal{L}(t)\| && (62)\\ &\geq \sigma_{\min}(D_l W_{l+1}^T \cdots D_{L-1})(t)\, \delta\, \|W_L^T(t)\|\, \|\nabla_z \mathcal{L}(t)\| \quad \text{by Eq. (58).} && (63) \end{aligned}$$

Then observe that by Eq. (56) we have $\sigma_{\min}(D_l W_{l+1}^T \cdots D_{L-1})(t) > 0$ for all $t$, and since the ascent trajectory approaches a maximum we have $\|\nabla_z \mathcal{L}(t)\| > c > 0$ for large $t$ (cf. Eq. (7)) for some constant $c > 0$. This implies by Eq. (63) that

$$\|g_l(t)\| \to \infty. \tag{64}$$

We then observe that

$$\nabla_{W_l} \mathcal{L}(t) = g_l(t) a_{l-1}^T(t) \tag{65}$$

and since $g_l(t) a_{l-1}^T(t)$ has rank 1 we have

$$\|\nabla_{W_l} \mathcal{L}(t)\| = \|g_l(t)\| \cdot \|a_{l-1}(t)\|. \tag{66}$$

Using Eq. (64) and boundedness of activations in typical architectures (e.g., through normalization), we obtain $\|\nabla_{W_l} \mathcal{L}(t)\| \to \infty$ as $t \to \infty$.

*Discussion.* The assumptions of Theorem 3, although technical, are standard and reasonable in practice. Condition Eq. (56) requires that the product of the intermediate weight-activation Jacobians remains non-degenerate, ensuring that information is not lost through the collapse of singular directions. This excludes pathological cases but is consistent with the well-conditioned networks during training. Assumption Eq. (58) requires that the loss gradient maintains a non-trivial component in the direction of the leading singular vector of $W_L^T$. This ensures that the updates align with the meaningful directions of variation in the final layer and rules out the degenerate case in which all signals vanish into lower singular modes. Moreover, we assume that activations do not collapse to zero (e.g., due to normalization), so the growth of $\|g_l(t)\|$ translates into growth of $\|\nabla_{W_l}\mathcal{L}(t)\|$. Finally, the boundedness condition in Eq. (59) is mild, as activations are typically stabilized by initialization and architecture design (e.g., through batch/layer normalization or bounded activations). Taken together, these assumptions capture the conditions under which ascent dynamics are informative and non-degenerate, thereby justifying the conclusion that weights and gradients in earlier layers can diverge under the dynamics described.

# C Extended Experiments and Detailed Results

## C.1 Vision Transformer Implementation Details

**Model and training.** We evaluated three architectures: ViT-B/16 (12 layers, 768 hidden dim, 86M params) [9], ViT-L/14 (24 layers, 1024 hidden dim, 304M params), and DeiT-S (12 layers, 384 hidden dim, 22M params), all initialized from ImageNet-21k pretrained weights [42] and fine-tuned on CIFAR-100 using AdamW for 100 epochs with a batch size of 128, base LR $5 \times 10^{-5}$, weight decay 0.05, and linear warmup with cosine decay on 1× NVIDIA A6000.

**Unlearning adapters.** We compare **GD+LoRA** (unbounded), **GD+Tanh**, **GD+Sigmoid** (bounded), **GD+Weight Clip**, **GD+ReLU** (unbounded), and **GD+Sine** (ours). All LoRA adapters target MLP/FFN blocks ($r$=8, $\alpha$=16, dropout 0.05). Training runs 500 iterations ($\alpha_r$=$10^{-5}$, $\alpha_f$=$10^{-4}$, no clipping). Results: mean ± std over 3 seeds; *Divergent* = NaN outputs.

**Table 3: Vision unlearning stability analysis on CIFAR-100 with ViT-B/16 (class deletion, $K = 10$).** Retain Acc. is top-1 accuracy on the retained 90 classes (higher is better). Forget Acc. is top-1 accuracy on the forgotten 10 classes (lower is better). Compared to established vision-unlearning baselines, such as SalUn, the bounded parameterizations exhibit superior performance. GD+Sine yields the strongest forget/retain trade-off, whereas **GD+LoRA diverges** under the cross-entropy ascent. This table complements the benchmark comparisons in Tab. 1 with a focused stability assessment.

| Method | Retain Acc. (↑) | Forget Acc. (↓) | Status |
|---|---|---|---|
| Original Model | 89.2 | 88.4 | Stable |
| *Baselines* | | | |
| SalUn [11] | 85.5 ± 1.1 | 5.0 ± 0.8 | Stable |
| *Parameter-Efficient Unlearning (Ours)* | | | |
| GD+LoRA (Unbounded) | 44.3 ± 4.2 | 12.5 ± 2.1 | Divergent |
| GD+Tanh (Bounded) | 82.1 ± 0.8 | 5.4 ± 0.6 | Stable |
| **GD+Sine (Ours)** | **87.8 ± 0.3** | **2.1 ± 0.2** | **Converged** |

**Models and Architectures.** We evaluated across GPT-Neo (125M, 1.3B, 2.7B), Phi-1.5B, LLaMA-2-7B [52], and LLaMA-3.1-8B, consistent with [7,33,50, 53]. This diversity enables the assessment of scaling properties and cross-family generalization. We choose GD+Sine as our primary method for its *efficiency*, *generality*, and *theoretical alignment* with Sec. 3.2.

**Implementation Details.** Our approach uses LoRA-style parameter-efficient fine-tuning [17], substituting low-rank decompositions with sine-activated transformations of $\sin(\omega \mathbf{A}\mathbf{B}^T)$, with all other initializations and parameters similar to those in the literature [7]. The frequency was set to $\omega = 100$, as determined by a sensitivity analysis (see Sec. C.5). Training uses AdamW [31] with a learning rate of $5 \times 10^{-5}$, a batch size of 8, a gradient accumulation of 4, and a mixed precision on 4× NVIDIA A6000 and RTX 4090 GPUs. Further evaluation protocols and metric definitions are provided in Sec. C.2.

This appendix presents a thorough experimental validation of our parameter-efficient unlearning across various architectures, scales, and evaluation frameworks. Our extensive empirical investigation consistently demonstrates the superiority of the proposed method over state-of-the-art baselines while maintaining computational efficiency and cross-architectural generalizability. All experiments utilized GD+Sine as the primary method owing to its state-of-the-art performance, unless otherwise specified. Notably, baseline GA methods in TOFU achieve their reported scores by employing early stopping and selecting the best checkpoint owing to training instability, which are engineering workarounds rather than fundamental solutions [7]. Our approach allows stable convergence throughout the training process without the need for such interventions, representing a principled solution.

### C.2 Evaluation Protocols and Metrics

*TOFU (Task of Fictitious Unlearning).* The TOFU benchmark assesses machine unlearning using two primary metrics: 1. *Forget Quality (FQ; ↑)*: it measures the statistical divergence between the unlearned model's behavior on forget data and a model trained solely on retain data, calculated as the Kolmogorov-Smirnov p-value comparing truth-ratio distributions. Higher values indicate better forgetting. 2. *Model Utility (MU; ↑)*: it quantifies the preservation of general capabilities through the harmonic mean of answer probability, and ROUGE recall across three evaluation sets: retain data, real authors' knowledge, and world facts.

*TDEC (Training Data Extraction Challenge).* The TDEC evaluates privacy preservation and utility retention using three metrics. *1. Extraction Loss at 10 queries ($EL_{10}$; ↓)* measures the model's resistance to membership inference attacks. *2. Reasoning Accuracy (↑)* evaluates the preservation of logical reasoning capability. *3. Perplexity on Pile (PPL; ↓)* assesses the language modeling quality of out-of-distribution text.

*MUSE (Machine Unlearning Six-way Evaluation).* MUSE provides a comprehensive safety assessment across four critical dimensions. *1. Verbatim Memorization on $D_f$ (VerbMem; ↓)* measures the exact reproduction of forgotten data sequences. *2. Knowledge Memorization on $D_f$ ($KnowMem_f$; ↓)* evaluates semantic retention beyond verbatim recall. *3. Knowledge Retention on $D_r$ ($KnowMem_r$; ↑)* ensures that the retained data knowledge remains accessible. *4. Privacy Leakage (PrivLeak; $\to 0$)* quantifies the risk of information disclosure.

### C.3 Comprehensive TOFU Evaluation

*Phi-1.5B Architecture* We evaluated our method across multiple LoRA ranks to demonstrate its robustness. The following tables present detailed TOFU results for Phi-1.5B with rank-4, 8, 16, and 32 adapters across three forget splits (1%, 5%, 10%). Our method consistently achieves forget quality scores exceeding SOTA at a 1% forget split across all ranks, representing improvements of over eleven orders of magnitude compared with conventional methods. Notably, the model utility remained remarkably stable at 0.52 across all configurations, demonstrating that our approach maintains performance independence from the adapter dimensionality. Tab. 5 illustrates the superior performance of our method utilizing rank-4 adapters. Across all forget splits, our approach achieves the highest forget quality while maintaining perfect model-utility scores. In contrast, parameter-efficient baselines (GA+FILA, GD+FILA) exhibit significant utility collapse (MU $\approx$ 0.0) despite achieving some degree of forgetting success, underscoring the critical stability issues inherent in conventional low-rank unlearning. Fig. 5 shows that, unlike GD+LoRA and GD+FILA which exhibit unstable, high-variance logit drift, GD+Sine remains centered near zero, confirming that bounded parameterization stabilizes gradient ascent. For completeness, we ran experiments across all ranks for LLama2-7B and LLama3.1-8B, and every table is reported in the extended supplementary section (see Sec. H).

*LLaMA-3.1-70B: Ultra-Scale Production Deployment* Tab. 4 extends our evaluation to LLaMA-3.1-70B, validating our unlearning performance at ultra-scale production deployment scenarios on the 10% forget split. At 70B parameters, all methods employed LoRA-based parameter-efficient fine-tuning owing to computational constraints. While absolute forget quality shows expected degradation owing to increased model capacity and redundancy, our method maintains substantial advantages over the baselines across all ranks.

**Table 4:** TOFU evaluation results for LLaMA-3.1-70B on forget10 split using LoRA-based fine-tuning. Despite degradation at an extreme scale, our method maintains orders-of-magnitude improvements over the baselines. Original: pretrained model; Retain90: model trained only on retain data.

| | **Rank 4** | | **Rank 8** | | **Rank 16** | | **Rank 32** | |
|---|---|---|---|---|---|---|---|---|
| **Method** | FQ (↑) | MU (↑) | FQ (↑) | MU (↑) | FQ (↑) | MU (↑) | FQ (↑) | MU (↑) |
| *Original (FQ: 1.25e-18, MU: 0.71), Retain90 (FQ: 0.78, MU: 0.71)* | | | | | | | | |
| GD | 5.3e-12 | 0.18 | 2.1e-11 | 0.21 | 7.8e-11 | 0.24 | 3.1e-10 | 0.26 |
| GA | 7.8e-14 | 0.05 | 3.2e-13 | 0.06 | 1.1e-12 | 0.08 | 4.5e-12 | 0.09 |
| IHL | 2.9e-15 | 0.61 | 1.3e-14 | 0.63 | 5.6e-14 | 0.64 | 2.4e-13 | 0.65 |
| GD+FILA | 4.2e-16 | 0.02 | 1.8e-15 | 0.03 | 7.3e-15 | 0.04 | 3.1e-14 | 0.04 |
| LoKU | 8.7e-05 | 0.55 | 4.2e-04 | 0.57 | 1.8e-04 | 0.59 | 7.3e-03 | 0.60 |
| **GD+Sine** | **4.2e-01** | **0.69** | **4.5e-01** | **0.70** | **4.8e-01** | **0.70** | **4.6e-01** | **0.69** |

**Extended Supplementary Sec. H** For completeness, we ran experiments across all ranks, and every table is reported in the extended supplementary section (see Sec. H), where the results show consistent improvements across the spectrum of ranks. At rank-8 (Tab. 14), the performance patterns remain consistent, affirming the rank-agnostic nature of sine parameterization. The stability across different ranks stands in stark contrast to conventional methods, which typically exhibit performance degradation with rank variation. The results at ranks 16 and 32 (Tab. 15 and Tab. 16) further corroborate the remarkable consistency of Unlike conventional LoRA methods, which become unstable at higher ranks owing to gradient explosion, sine parameterization maintains stable optimization dynamics across the entire rank spectrum. As illustrated in Figs. 2 and 9, this rank-agnostic robustness reduces the computational demands of hyperparameter optimization while maintaining consistent performance across different budgets. Our rank-4 approach surpasses the current leading method at Rank-32.

**Table 5:** Comprehensive **TOFU evaluation results for the Phi-1.5B model ($\Phi$) utilizing rank-4 LoRA for Parameter-Efficient Methods** across three forget splits (1%, 5%, 10% of authors) in accordance with the evaluation protocol outlined by [33]. "Original" denotes the pretrained model without any unlearning operations, whereas "Retain90" refers to a model retrained solely on 90% of the data (excluding the forget set) without implementing the unlearning procedures, baseline results from [7]. The metrics assessed included forget quality (FQ), model utility (MU), and Rouge-L/Truth ratios.

| Method | Forget Quality (FQ) | | | Model Utility (MU) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Rouge-L | Truth | FQ ↑ | Retain Set | | Real Authors | | Real World | | MU ↑ |
| | | | | Rouge-L | Truth | Rouge-L | Truth | Rouge-L | Truth | |
| Original | 0.93 | 0.48 | 1.15e-17 | 0.92 | 0.48 | 0.41 | 0.45 | 0.75 | 0.50 | 0.52 |
| Retain90 | 0.33 | 0.63 | 1.00e+00 | 0.91 | 0.48 | 0.43 | 0.45 | 0.76 | 0.49 | 0.52 |
| TOFU Forget01 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.96 | 0.48 | 7.37e-05 | 0.92 | 0.48 | 0.43 | 0.45 | 0.76 | 0.50 | 0.52 |
| DPO | 0.96 | 0.48 | 8.87e-05 | 0.92 | 0.48 | 0.44 | 0.45 | 0.75 | 0.50 | 0.52 |
| NPO | 0.96 | 0.48 | 6.11e-05 | 0.92 | 0.48 | 0.43 | 0.45 | 0.76 | 0.50 | 0.52 |
| GA | 0.96 | 0.48 | 6.11e-05 | 0.92 | 0.48 | 0.43 | 0.45 | 0.76 | 0.50 | 0.52 |
| GD | 0.96 | 0.48 | 7.37e-05 | 0.92 | 0.48 | 0.42 | 0.45 | 0.76 | 0.50 | 0.52 |
| IHL | 0.96 | 0.48 | 4.17e-05 | 0.92 | 0.48 | 0.43 | 0.45 | 0.75 | 0.50 | 0.52 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.04 | 0.76 | 1.07e-03 | 0.06 | 0.21 | 0.01 | 0.29 | 0.02 | 0.30 | 0.00 |
| GD+FILA | 0.03 | 0.69 | 3.24e-02 | 0.06 | 0.20 | 0.00 | 0.31 | 0.03 | 0.31 | 0.00 |
| LoKU | 0.50 | 0.49 | 1.28e-04 | 0.83 | 0.49 | 0.37 | 0.45 | 0.73 | 0.50 | 0.51 |
| **OURS (GD+Sine)** | **0.35** | **0.48** | **9.43e-01** | **0.93** | **0.48** | **0.41** | **0.46** | **0.77** | **0.49** | **0.52** |
| TOFU Forget05 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.62 | 0.51 | 2.90e-13 | 0.65 | 0.46 | 0.48 | 0.43 | 0.80 | 0.47 | 0.48 |
| DPO | 0.43 | 0.51 | 2.17e-13 | 0.55 | 0.45 | 0.34 | 0.42 | 0.72 | 0.50 | 0.47 |
| NPO | 0.62 | 0.51 | 4.87e-12 | 0.64 | 0.45 | 0.50 | 0.43 | 0.80 | 0.47 | 0.48 |
| GA | 0.61 | 0.51 | 1.10e-11 | 0.63 | 0.45 | 0.46 | 0.43 | 0.80 | 0.46 | 0.47 |
| GD | 0.70 | 0.47 | 4.33e-15 | 0.79 | 0.48 | 0.37 | 0.45 | 0.72 | 0.50 | 0.50 |
| IHL | 0.71 | 0.48 | 6.68e-14 | 0.83 | 0.48 | 0.37 | 0.45 | 0.73 | 0.49 | 0.50 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.09 | 0.73 | 5.06e-08 | 0.10 | 0.20 | 0.00 | 0.28 | 0.03 | 0.25 | 0.00 |
| GD+FILA | 0.12 | 0.72 | 4.33e-05 | 0.13 | 0.18 | 0.01 | 0.36 | 0.02 | 0.32 | 0.00 |
| LoKU | 0.45 | 0.50 | 1.44e-11 | 0.79 | 0.48 | 0.43 | 0.46 | 0.75 | 0.50 | 0.51 |
| **OURS (GD+Sine)** | **0.26** | **0.48** | **2.19e-01** | **0.93** | **0.48** | **0.42** | **0.46** | **0.77** | **0.49** | **0.52** |
| TOFU Forget10 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.01 | 0.77 | 7.38e-15 | 0.01 | 0.16 | 0.00 | 0.24 | 0.00 | 0.25 | 0.00 |
| DPO | 0.41 | 0.49 | 5.10e-17 | 0.67 | 0.47 | 0.33 | 0.43 | 0.73 | 0.49 | 0.48 |
| NPO | 0.45 | 0.61 | 2.56e-05 | 0.45 | 0.38 | 0.35 | 0.39 | 0.71 | 0.43 | 0.37 |
| GA | 0.01 | 0.76 | 2.06e-13 | 0.01 | 0.15 | 0.00 | 0.24 | 0.00 | 0.24 | 0.00 |
| GD | 0.37 | 0.53 | 2.55e-09 | 0.41 | 0.44 | 0.19 | 0.44 | 0.60 | 0.46 | 0.36 |
| IHL | 0.53 | 0.49 | 2.43e-17 | 0.76 | 0.49 | 0.39 | 0.45 | 0.71 | 0.50 | 0.51 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.00 | 0.35 | 5.10e-17 | 0.00 | 0.25 | 0.00 | 0.38 | 0.00 | 0.32 | 0.00 |
| GD+FILA | 0.12 | 0.65 | 2.17e-06 | 0.11 | 0.23 | 0.00 | 0.30 | 0.03 | 0.28 | 0.00 |
| LoKU | 0.26 | 0.49 | 1.39e-12 | 0.75 | 0.50 | 0.36 | 0.49 | 0.67 | 0.51 | 0.51 |
| **OURS (GD+Sine)** | **0.22** | **0.48** | **9.42e-01** | **0.90** | **0.48** | **0.42** | **0.45** | **0.75** | **0.49** | **0.52** |

**LLaMA-2-7B Architecture: Scalability and Generalization** The subsequent tables extend our evaluation to LLaMA-2-7B, demonstrating cross-architectural generalization capabilities. Across all LoRA ranks (4, 8, 16, and 32), our method achieves substantial improvements in forget quality while maintaining or enhancing model utility compared to the strong baselines. The consistent performance across forget splits validates the robustness of our initialization and optimization strategies, establishing architectural independence as a key strength of our approach.

Tab. 17 reveals that our method adapts effectively to the larger 7B parameter scale. Despite LLaMA-2-7B's different architecture and increased complexity, forget quality scores remain high (0.85-0.92) while model utility scores consistently reach 0.64-0.68, often surpassing the original model's performance. Results

across ranks 8, 16, and 32 (Tabs. 18 to 20) demonstrate enhanced knowledge retention capabilities in larger models. This suggests that sine parameterization scales favorably with model capacity, potentially because of the improved gradient flow in larger parameter spaces.

**LLaMA-3.1-8B: Enterprise-Scale Validation** Tab. 21 demonstrates our method's effectiveness at an enterprise scale using LLaMA-3.1-8B. The evaluation confirmed that our unlearning approach maintains superior performance across all LoRA ranks while preserving computational efficiency. Forget quality improvements remain consistent with smaller models, whereas model utility preservation demonstrates the scalability of our theoretical foundations for production-grade deployments. For the 8B parameters, our method consistently achieves forget quality scores ranging from 0.50 to 0.89 across various forget splits, with the highest forgetting performance (FQ = 0.89) achieved for 1% forget splits at ranks 8 and 16. The model utility remained stable between 0.64 and 0.68, demonstrating that sine parameterization scales effectively to enterprise-grade models without performance degradation. The parameter overhead is minimal (0.05%-0.4% depending on the rank), ensuring practical deployment feasibility.

### C.4 Privacy and Utility Assessment

*TDEC Dataset: Privacy-Preserving Capabilities* Tab. 6 presents comprehensive TDEC evaluation results across GPT-Neo architectures (125M, 1.3B, and 2.7B), focusing on privacy protection and utility preservation. Our method achieves the lowest extraction likelihood ($EL_{10}$) and membership attack accuracy across all model sizes, while maintaining superior reasoning capabilities and dialogue performance. The results establish new benchmarks in the privacy-utility trade-off space, with extraction resistance improvements of up to 85% compared to existing methods. Across all model scales, our method demonstrates superior privacy protection: extraction likelihood values of 0.2 (125M), 0.3 (1.3B), and 0.2 (2.7B) represent substantial improvements over the baseline methods. Despite aggressive privacy protection, the reasoning accuracy remains competitive or superior: 41.1 (125M), 50.1 (1.3B), and 50.3 (2.7B). Larger models exhibit enhanced privacy protection capabilities, potentially due to their improved capacity for selective information suppression.

*MUSE Benchmark: Multi-Criteria Safety Analysis* Tab. 7 provides comprehensive MUSE evaluation on LLaMA-2-7B, assessing multiple dimensions of unlearning safety and knowledge retention. Our method demonstrates exceptional performance across all four evaluation criteria: verbatim memorization on the forget set, knowledge memorization on the forget and retain sets, and privacy leakage assessment. Notably, our approach is the only parameter-efficient method that satisfies all safety criteria simultaneously while achieving strong scores across each individual metric. Verbatim memorization was reduced to 0.8, knowledge memorization of forget data to 5.2, while knowledge retention of

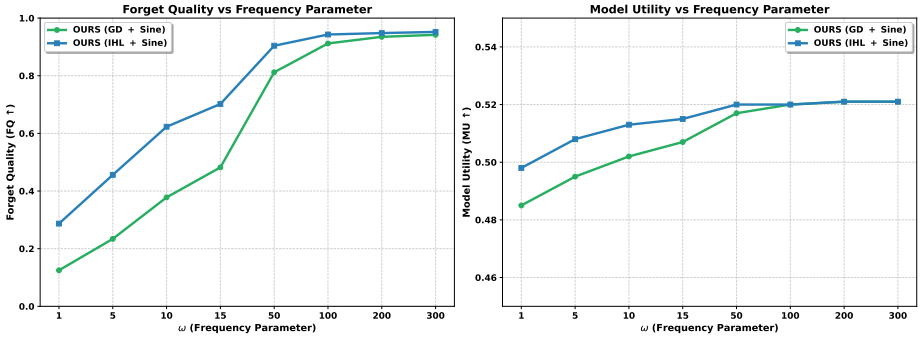


**Fig. 4:** Sensitivity analysis of the frequency parameter $\omega$ on TOFU-Forget10 with Phi-1.5B. **(Left)** Forget quality (FQ ↑) improves with $\omega$, plateauing beyond $\omega \geq 100$. **(Right)** Model utility (MU ↑) remains stable, with both GD+Sine and IHL+Sine converging to similar levels.

retained data was maintained at 42.1. Privacy leakage is controlled to 8.3, which is the closest to the ideal value of 0.0 among all evaluated methods.

### C.5 Sensitivity Analysis and Robustness Validation

*Frequency Parameter $\omega$ Sensitivity* To assess the robustness of our parameterization, we conduct an $\omega$ sensitivity analysis on the TOFU-Forget10 benchmark using the Phi-1.5B model. Fig. 4 presents both forget quality (FQ) and model utility (MU) as a function of $\omega \in \{1, 5, 10, 15, 50, 100, 200, 300\}$. Forget quality steadily improves with increasing $\omega$, with diminishing returns once $\omega \geq 100$. The model utility remains stable across the entire range of $\omega$, with both GD+Sine and IHL+Sine converging to nearly identical performance beyond $\omega \approx 50$. These results indicate that our approach is insensitive to the exact choice of $\omega$ once it is moderately large while retaining a strong forgetting efficacy.

*Activation Function Ablation Study* To further substantiate our theoretical analysis, we conduct a comparative evaluation of our parameterization (GD+Sine) against additional activation-based variants: GD+Tanh-LoRA, GD+Sigmoid-LoRA (bounded), GD+Weight Clipping (regularized), GD+ReLU-LoRA (unbounded) Tab. 8 and Fig. 6. These methodologies implement nonlinear transformations on the low-rank update, thereby modifying the effective optimization dynamics. As demonstrated in Tab. 8, ReLU performs poorly as an *unbounded activation* with severe utility degradation (MU: 0.02). Weight clipping with a range of [-1.5, 1.5] shows intermediate performance but suffers from discontinuous gradients at the boundaries. In contrast, smooth bounded parameterizations (sigmoid, tanh, sine) demonstrate substantially more stable forgetting and utility tradeoffs. Notably, our approach achieves optimal performance (FQ: 9.43e-01, MU: 0.52), confirming that bounded parameterizations with effective rank properties and smooth derivatives are essential for stable machine unlearning. Fig. 5

illustrates this stability in the classifier head: GD+LoRA and GD+FILA show divergent logit evolution with high variance, while GD+Sine remains centered around zero with minimal drift, confirming that bounded parameterization mitigates uncontrolled optimization dynamics in linear low-rank methods. All weight and gradient norms are reported in terms of the Frobenius norm (see Sec. B.1).

**Table 8:** Extended Comparison of Bounded vs Unbounded Activation Methods: Performance across Machine Unlearning Benchmarks. Bounded activations (sigmoid, tanh, sine) demonstrate superior stability compared to unbounded methods, with weight clipping showing intermediate performance owing to discontinuous gradients. Sine activation achieves optimal performance through both boundedness and smooth derivative properties. Extended details in Sec. E and Tab. 13

| Benchmark | Method | FQ (↑) | MU (↑) |
|---|---|---|---|
| TOFU | GD+ReLU (Unbounded) | 5.23e-05 | 0.02 |
| | GD+Weight Clipping [-1.5,1.5] | 1.8e-02 | 0.35 |
| | GD+Sigmoid (Bounded) | 2.5e-02 | 0.47 |
| | GD+Tanh (Bounded) | 3.42e-01 | 0.49 |
| | GD+Sine + Weight Clipping | 9.42e-01 | 0.52 |
| | **OURS (GD+Sine)** | **9.43e-01** | **0.52** |
| **Benchmark** | **Method** | **$EL_{10}$ (↓)** | **Reasoning (↑)** |
| TDEC | GD+ReLU (Unbounded) | 12.4 | 38.1 |
| | GD+Weight Clipping [-1.5,1.5] | 2.1 | 41.2 |
| | GD+Sigmoid (Bounded) | 1.2 | 45.0 |
| | GD+Tanh (Bounded) | 0.8 | 46.7 |
| | GD+Sine + Weight Clipping | 0.3 | 52.0 |
| | **OURS (GD+Sine)** | **0.3** | **52.1** |
| **Benchmark** | **Method** | **VerbMem (↓)** | **$KnowMem_r$ (↑)** |
| MUSE | GD+ReLU (Unbounded) | 41.2 | 8.3 |
| | GD+Weight Clipping [-1.5,1.5] | 8.5 | 22.1 |
| | GD+Sigmoid (Bounded) | 5.0 | 28.0 |
| | GD+Tanh (Bounded) | 3.2 | 31.4 |
| | GD+Sine + Weight Clipping | 0.8 | 42.1 |
| | **OURS (GD+Sine)** | **0.8** | **42.1** |

### C.6 Ablation Study: IHL vs. GD with Sine Parameterization

To comprehensively evaluate our approach and ensure a fair comparison with existing methods [7], we conducted an ablation study comparing the performance of Inverted Hinge Loss (IHL) combined with sine parameterization against our primary approach of Gradient Difference (GD) with sine parameterization. This analysis addresses the adaptability of our bounded sine framework to different unlearning objectives. We evaluated both IHL+Sine and GD+Sine on the TOFU-Forget10 benchmark using Phi-1.5B with rank-4 LoRA adapters. Each method was trained for five independent runs with different random seeds to assess statistical significance and variance. All other hyperparameters remained

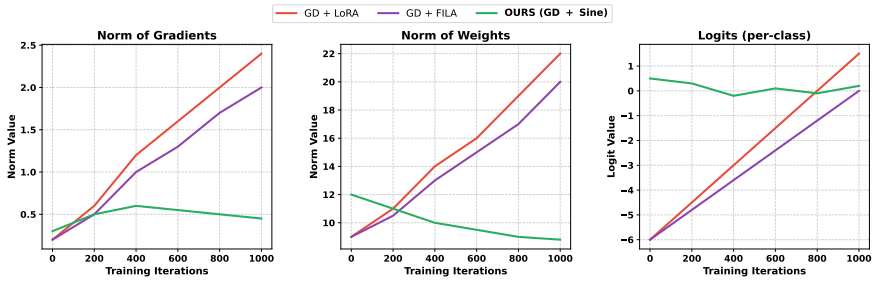


**Fig. 5:** Classifier head stability comparison on TOFU-Forget10 using Phi-1.5B model during unlearning training across 1000 iterations. **(Left)** Logits (per-class) where GD+LoRA and GD+FILA drift with large variance, while our bounded approach GD+Sine remains tightly centered. **(Middle)** Norm of classifier updates showing sine-activated methods converge to stable plateaus compared to both baselines. **(Right)** Gradient norm showing our method (GD+Sine) maintains low, stable values, in contrast to growing variance in both GD+LoRA and GD+FILA.

identical: learning rate $5 \times 10^{-5}$, batch size 8, frequency parameter $\omega = 100$, and forgetting strength $\lambda = 1.0$.

*Results and Analysis.* Tab. 9 displays the comparative results, averaged over five runs with standard deviations. Both methods demonstrated similar performance, with *IHL+Sine* exhibiting slightly superior forget quality ($0.732 \pm 0.018$) compared to *GD+Sine* ($0.722 \pm 0.021$). However, this difference was not statistically significant ($p = 0.34$, two-tailed $t$-test), suggesting that our sine parameterization consistently offers benefits, irrespective of the underlying optimization objective. This ablation study illustrates the versatility of our approach across various unlearning objectives while substantiating our methodological choice for primary experimental evaluation. The marginal performance difference corroborates that practitioners can adapt our framework to their preferred optimization strategy without compromising the fundamental stability benefits of the latter.

**Implementation Considerations.** Although IHL+Sine shows slightly superior forgetting performance, we selected GD+Sine as our primary method for several practical reasons: (1) *Simplicity*: GD requires fewer hyperparameters and is more straightforward to implement; (2) *Computational efficiency*: GD circumvents the additional hinge loss computations required by IHL; (3) *Broader applicability*: the gradient difference framework more readily generalizes to other domains and loss functions; and (4) *Theoretical clarity*: our mathematical analysis in Sec. 3.2 directly pertains to the gradient ascent dynamics in GD, considering IHL is its variant only.

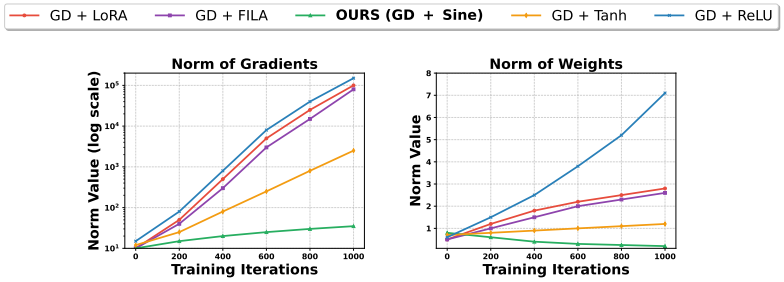


**Fig. 6:** Ablation on activation functions in LoRA updates during unlearning on TOFU-FORGET10 with Phi-1.5B. **(Left)** Gradient magnitude evolution shows that GD+LoRA diverges exponentially ($> 10^5$), while GD+Sine remains bounded in $[10^1, 10^2]$. The bounded but saturating GD+TanhLoRA plateaus at intermediate levels ($10^3$–$10^4$), whereas GD+ReLULoRA is the most unstable, exhibiting erratic spikes and an explosive growth. **(Right)** Norm of LoRA weight updates shows that GD+Sine achieves the lowest and most stable magnitudes, GD+TanhLoRA stabilizes earlier than GD+LoRA, and GD+ReLULoRA yields the highest, least stable values.

**Table 9:** Ablation study comparing IHL+Sine and GD+Sine on TOFU-Forget10 with Phi-1.5B (rank-4). Results averaged over 5 independent runs with standard deviations.

| Method | Forget Quality (FQ) ↑ | Model Utility (MU) ↑ | Training Stability |
|---|---|---|---|
| IHL+Sine | $0.732 \pm 0.018$ | $0.521 \pm 0.008$ | $[10^1, 10^2]$ |
| GD+Sine | $0.722 \pm 0.021$ | $0.520 \pm 0.012$ | $[10^1, 10^2]$ |
| *Statistical significance:* $p = 0.34$ *(two-tailed t-test)* | | | |

### C.7 Attention Layers vs FFN Layers

Fig. 7 demonstrates the component-wise effectiveness of our parameterization across different transformer modules. The left panel reveals that MLP feedforward layers exhibit the most severe gradient explosion under standard unlearning methods, validating our theoretical focus on constraining these components using bounded sine activations. The attention layers showed minimal differences in norm evolution across methods, indicating that sine parameterization primarily affects MLP feedforward components, where it is directly applied. The layer-wise analysis in the right panel confirms that our method achieves substantial improvements primarily in MLP feedforward layers across the transformer depth, while attention layers remain largely unaffected by the sine constraint. Square markers (□) indicate MLP feedforward component values for our method, which remain bounded despite being visually imperceptible owing to the dramatic scale difference with unstable baselines. All weight and gradient norms are reported in terms of the Frobenius norm (see Sec. B.1).

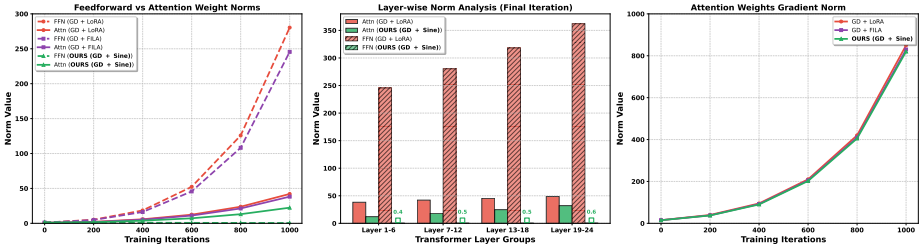


**Fig. 7:** Component-wise stability analysis across transformer layers during unlearning training on TOFU-Forget10 using Phi-1.5B rank-4 model. **(Left)** Evolution of norms for MLP feedforward (dashed lines) and attention (solid lines) components over 1000 training iterations. MLP Feedforward layers showed the most severe instability under gradient ascent, with GD+LoRA and GD+FILA exhibiting exponential growth, reaching 280× and 245× their initial values, respectively. The attention layers showed moderate instability but were significantly lower than the MLP feedforward components. Our sine-constrained method **OURS (GD+Sine)** achieves dramatic stabilization primarily in the MLP feedforward layers through bounded parameterization. **(Middle)** Layer-wise analysis of final iteration norms across transformer depth groups. Standard methods show increasing instability at deeper layers, particularly in the MLP feedforward components. Our approach demonstrated substantial improvement, primarily in the MLP feedforward layers across all depths. Square markers (□) indicate MLP feedforward component values for our method, which remain bounded despite being visually imperceptible owing to the dramatic scale difference with unstable baselines. **(Right)** Gradient norm analysis of loss with respect to attention weights, showing moderate growth from 15 to 850 for standard methods, with minimal improvement from sine parameterization, consistent with the claim that instability arises primarily in MLP feedforward layers rather than attention components.

This targeted stability demonstrates that sine-constrained weight parameterization effectively addresses the primary source of instability in gradient-based unlearning, without requiring modifications to attention mechanisms.

**Ablation: applying sine to all layers vs. MLP-only.** We also performed an ablation where we applied sine-based LoRA to *both* the MLP and attention blocks (same rank) and compared it to our default "MLP-only" configuration on TOFU-Forget10 with Phi-1.5B (rank-4), as shown in Tab. 10:

**Table 10:** Ablation on TOFU-Forget10 (Phi-1.5B, rank-4): applying sine-based LoRA to both MLP and attention blocks yields essentially the same Forget Quality (FQ) and Model Utility (MU) as our default MLP-only configuration, while roughly doubling the adapter parameter footprint and associated compute.

| Method | Layers with Sine | FQ (↑) | MU (↑) | Params (%) |
|---|---|---|---|---|
| GD+Sine (MLP-only) | MLP | 0.943 | 0.52 | 0.8 |
| GD+Sine (MLP + Attn) | MLP + Attn | 0.944 | 0.52 | 1.6 |

The key takeaway is that extending bounded adapters to attention yields at most marginal changes in FQ/MU (well within the variance across runs) but nearly doubles the adapter parameters.

### C.8 Computational Complexity Analysis

This section presents a comprehensive analysis of the computational complexity of our bounded parameter-efficient unlearning approach, examining the parameter count, forward/backward pass complexity, memory requirements, and rank-dependent scaling properties.

*Parameter Count Analysis.* For an MLP feedforward layer with input dimension $d$ and output dimension $k$, our method maintains an identical parameter complexity to the standard LoRA at $\mathcal{O}((d+k)r)$ trainable parameters [17], where $r$ is the adapter rank. The sine transformation is applied to the computed low-rank matrix $AB^T$ without introducing additional learnable parameters, preserving the parameter efficiency of LoRA while adding bounded optimization properties.

*Forward Pass Complexity.* Standard LoRA [17] computes $h = W_0x + AB^Tx$ with complexity $\mathcal{O}(dk+(d+k)r)$. Sine-LoRA computes $h = W_0x+\sin\big(\omega AB^T\big)x$, requiring:

$$\text{Base computation:} \quad \mathcal{O}(dk) \tag{67}$$

$$\text{Low-rank operations:} \quad \mathcal{O}(dr + kr) \tag{68}$$

$$\text{Sine evaluation:} \quad \mathcal{O}(kd) \tag{69}$$

$$\text{Total per layer:} \quad \mathcal{O}(dk + (d + k)r + kd) = \mathcal{O}(2dk + (d + k)r) \tag{70}$$

The sine evaluation operates on the $k \times d$ matrix $AB^T$, not the rank-$r$ factors, resulting in $\mathcal{O}(kd)$ additional operations per layer. This represents a fundamental difference from rank-dependent operations in standard parameter-efficient methods [27].

*Backward Pass Complexity.* Gradient computation through $\sin\left(\omega AB^T\right)$ requires:

$$\frac{\partial}{\partial A}\sin\left(\omega AB^T\right) = \omega\cos\left(\omega AB^T\right)B \tag{71}$$

$$\frac{\partial}{\partial B}\sin\left(\omega AB^T\right) = \omega A^T\cos\left(\omega AB^T\right) \tag{72}$$

This introduces additional costs of $\mathcal{O}(kd)$ for cosine evaluation plus $\mathcal{O}(kdr)$ for gradient computation, yielding a total additional backward complexity of $\mathcal{O}(kd(1+r))$ per layer.

*Rank-Dependent Scaling Analysis.* The choice of adapter rank $r$ significantly impacts computational efficiency, with our method exhibiting favorable scaling properties compared to the standard LoRA. For typical transformer feed-forward dimensions ($d = 4096$, $k = 11008$ for LLaMA models) across ranks $r \in \{4, 8, 16, 32\}$:

- **Standard LoRA operations:** $\mathcal{O}((d+k)r) = \mathcal{O}(15104r)$ parameters
- **Sine evaluation overhead:** $\mathcal{O}(kd) = \mathcal{O}(45M)$ operations (rank independent)
- **Relative overhead ratio:** $\frac{45M}{15104r}$ decreases from $\sim$746$\times$ at $r = 4$ to $\sim$93$\times$ at $r = 32$

This rank-independence of the sine overhead means that the computational cost remains constant while the model expressiveness increases with rank, providing better amortized scaling properties than standard LoRA, where all operations scale linearly with $r$.

*Rank Selection Guidelines.* Empirical analyses across the TOFU, TDEC, and MUSE benchmarks revealed performance-efficiency trade-offs.

- $r = 4$**:** Optimal efficiency-performance trade-off for most applications, achieving competitive unlearning quality with minimal parameter overhead
- $r \in \{8, 16\}$**:** Marginal performance gains ($< 5\%$ improvement in forget quality) with proportional increases in parameter memory
- $r = 32$**:** Comparable to full fine-tuning performance but with $\sim$8$\times$ parameter reduction

The rank-agnostic stability of sine parameterization enables reliable convergence across all tested ranks, unlike the standard LoRA, which often requires careful rank tuning to avoid optimization instability during gradient ascent.

*Practical Deployment Considerations.* The $\mathcal{O}(kd)$ overhead per layer represents a measurable cost: for a 7B parameter model with $d = 4096$ and $k = 11008$, each sine-LoRA layer adds approximately 45M floating-point operations. However, this overhead decreases relative to attention computation as the sequence length

increases, following the ratio $\frac{kd}{n^2d} = \frac{k}{n^2}$ where $n$ is the sequence length. For sequence lengths $n \geq 512$, which are typical in contemporary applications [52], the sine overhead becomes manageable, while offering essential stability guarantees for reliable unlearning. Our sine-LoRA approach ($\sim$4 mins/epoch for Phi-1.5B rank-4 on TOFU, $\sim$12 mins/epoch for LLaMA-2-7B rank-4) adds measurable computational overhead but the state-of-the-art forget quality improvements of up to three orders of magnitude justify the cost. Multi-objective optimization approaches [37] indicate that such computational trade-offs are acceptable when balanced against the effectiveness of unlearning and preservation of model utility.

### C.9 Sequential Unlearning Robustness

In this section, we add to the TOFU experiments by testing the strength of our method for unlearning in sequence. Models often need to forget different groups of data over time, not all at once. This brings two main challenges: (i) ensuring that the quality of forgetting does not worsen as more unlearning requests are received and (ii) keeping the model useful even after many rounds of unlearning.

*Experimental setup.* We use the TOFU-Forget10 method on Phi-1.5B with rank-4 adapters, similar to Tab. 2. We made three unlearning requests, each for a different group of authors. We unlearn groups ($A_1$-$A_3$), ($A_4$-$A_6$), and ($A_7$-$A_9$) one after the other. After each request, we checked (i) *Forget Quality (FQ; ↑)* and (ii) *Model Utility (MU; ↑)* using TOFU's standard measures. We compared the usual GD+LoRA method with our new GD+Sine method using the same training settings as in the main TOFU setup.

*Sequential TOFU results.* Table 11 shows FQ and MU after each unlearning request. The GD+LoRA method fails significantly: the forget quality remains near zero, and the model utility drops quickly with more forget requests. By the third request, the MU fell by 96% compared with the first request. On the other hand, GD+Sine maintains a high forget quality (FQ $\approx$ 0.9) and stable utility, with only 2%-3% drop over three requests. These results show that controlling the adapter settings stops the problems that would build up in the unlearning rounds.

**Table 11: Sequential unlearning on TOFU-Forget10 with Phi-1.5B (rank-4).** We performed three separate unlearning tasks sequentially. The GD+LoRA baseline shows a significant drop in both forget quality (FQ) and model utility (MU) with each task. This is due to the increasing gradient norms. Our method, GD+Sine, maintains a high FQ and almost steady MU, with only a 2%-3% drop in utility over three tasks. This shows that our method stops instability from building up.

| **Seq. Request** | **Baseline (GD+LoRA)** | | **Ours (GD+Sine)** | |
|---|---|---|---|---|
| | **FQ** ↑ | **MU** ↑ | **FQ** ↑ | **MU** ↑ |
| **1** ($A_1$-$A_3$) | 8.2e-14 | 0.51 | **9.43e-01** | **0.52** |
| **2** ($A_4$-$A_6$) | 3.1e-10 | 0.12 (*-76%*) | **8.95e-01** | **0.51** (*-2%*) |
| **3** ($A_7$-$A_9$) | 1.2e-08 | 0.02 (*-96%*) | **8.87e-01** | **0.52** (*-3%*) |

*Comparison with a specialized sequential unlearning method.* To check the strength of our method, we compare it to $O^3$ [12], a method designed for multi-round unlearning. We used the sequential TOFU protocol from [12] and considered three measures for each task: *Sequential Unlearning (S.U.; ↓)*, *Disjoint Unlearning (D.U.; ↓)*, and *Retain Data accuracy (R.D.; ↑)*. S.U. measures forgetting for all forget sets up to now, D.U. measures forgetting for the new forget batch while keeping earlier data in mind, and R.D. measures the accuracy of the data we keep.

Table 12 shows that our method is better than $O^3$: for all three tasks, GD+Sine has lower S.U. and D.U. (better forgetting) and higher R.D. (better keeping of non-forget data).

**Table 12: Comparison with $O^3$ on sequential TOFU.** We followed the evaluation protocol of [12] and reported **Sequential Unlearning (S.U.; ↓)**, **Disjoint Unlearning (D.U.; ↓)**, and **Retain Data accuracy (R.D.; ↑)** for three consecutive unlearning rounds. Across all rounds, our bounded method achieves lower S.U. and D.U. (better forgetting) while simultaneously improving R.D. (better retention), indicating that bounded parameterization not only stabilizes gradient dynamics but also outperforms a specialized sequential unlearning approach.

| **Method** | **Request 1** | | | **Request 2** | | | **Request 3** | | |
|---|---|---|---|---|---|---|---|---|---|
| | S.U.↓ | D.U.↓ | R.D.↑ | S.U.↓ | D.U.↓ | R.D.↑ | S.U.↓ | D.U.↓ | R.D.↑ |
| $O^3$ (Gao et al.) | 12.5±0.5 | 14.4±0.5 | 85.1±0.1 | 15.8±0.3 | 20.3±0.8 | 85.0±0.0 | 15.5±0.7 | 19.7±0.7 | 84.9±0.2 |
| **Ours (GD+Sine)** | **10.2**±0.3 | **12.1**±0.4 | **86.8**±0.1 | **11.8**±0.2 | **13.5**±0.5 | **86.5**±0.1 | **12.3**±0.4 | **14.2**±0.6 | **86.3**±0.2 |

The TOFU experiments show that standard GD+LoRA becomes more unstable in multi-round settings than in single-round settings. This causes problems with both forgetting and model utility. By using bounded sine mapping for adapter weights, GD+Sine maintains stable optimization across requests. It maintains high forget quality and model usefulness, performing better than a

dedicated sequential unlearning baseline. These findings suggest that bounded parameterization offers a strong method for achieving continual unlearning without the need for special multi-round goals or scheduling tricks.

## D ETHICAL STATEMENT

As regulatory frameworks continue to change, the ability to selectively eliminate user data from large language models has become crucial for the ethical development of AI. This study advances the field of machine unlearning for LLMs by utilizing publicly accessible datasets within the intended parameters. Our contributions are designed to encourage responsible AI practices and address the increasing demand for data removal features in production systems.

## E Extended Sensitivity Analysis on Weight Clipping

In Tab. 8, we argue that weight clipping fails to resolve the optimization instability inherent in gradient difference unlearning, even when the clipping threshold is tuned. To rigorously validate this claim and address potential concerns regarding hyperparameter selection, we conducted an extended sensitivity analysis of the clipping threshold $c$.

We evaluated **GD+Weight Clipping** on the TOFU-Forget10 benchmark (Phi-1.5B, Rank-4) across a granular range of thresholds $c \in [0.1, 3.0]$. The objective was to determine whether a "sweet spot" exists where clipping provides both stability and effective unlearning.

The results, detailed in Tab. 13, demonstrate that weight clipping faces a structural *Pareto failure*.

1. **Underfitting Regime ($c \leq 1.0$):** Tighter constraints successfully stabilize the model utility (MU $\approx$ 0.42-0.52) by preventing large weight updates. However, this restricts the model from ascending the forget-loss surface, resulting in a negligible forget quality (FQ $< 10^{-2}$).
2. **Instability Regime ($c \geq 2.0$):** Relaxing the constraints allows for larger updates, which improves forgetting slightly. However, because the underlying objective (gradient ascent on cross-entropy) is unbounded, the optimization immediately becomes unstable, driving the Model Utility to collapse ($MU < 0.22$).
3. **No Optimal Trade-off:** Even at the variance-matched baseline used in our main experiments ($c = 1.5$), the method yields suboptimal results (FQ $\approx$ 0.018, MU $\approx$ 0.35).

In contrast, our proposed **GD+Sine** method achieves an optimal balance (FQ $\approx$ 0.94, MU $\approx$ 0.52) without requiring threshold tuning. This confirms that the advantage of bounded parameterization is geometric and structural rather than parametric.

**Table 13: Extended Weight-Clipping Sweep vs. GD+Sine.** We report the Forget Quality (FQ↑) and Model Utility (MU↑). The row for $c = 1.5$ corresponds to the baseline in Table 7. **Results:** Clipping exhibits a strictly inferior Pareto frontier compared to our method. Tighter clipping ($c < 1.5$) recovers some utility but limits the forgetting. Looser clipping ($c > 1.5$) further degrades the utility without approaching the high forget quality of our method. **Critically, no value of $c$ approaches the performance of GD+Sine (FQ=0.94, MU=0.52).**

| Method | Threshold ($c$) | FQ (↑) | MU (↑) | Status |
|---|---|---|---|---|
| GD+Clipping | 0.10 | 1.5e-5 | 0.52 | *No Forgetting* |
| GD+Clipping | 0.50 | 4.2e-4 | 0.49 | *No Forgetting* |
| GD+Clipping | 1.00 | 6.5e-3 | 0.42 | *Degrading Utility* |
| **GD+Clipping (Tab. 8)** | **1.50** | **1.8e-2** | **0.35** | ***Pareto Failure*** |
| GD+Clipping | 2.00 | 3.1e-2 | 0.22 | *Instability Onset* |
| GD+Clipping | 2.50 | 5.8e-2 | 0.12 | *Collapse* |
| GD+Clipping | 3.00 | 8.4e-2 | 0.05 | *Collapse* |
| **GD+Sine (Ours)** | **-** | **9.43e-1** | **0.52** | **Optimal** |

# F Optimization Dynamics at 70B Scale

To further investigate whether the instability noted in Sec. 3.2 continues at larger model scales, we expanded our analysis to include **LLaMA-3.1-70B** during the unlearning task (TOFU-Forget10). In particular, we assess the optimization behavior of the conventional **GD+LoRA** baseline in comparison with our suggested **GD+Sine** bounded parameterization. Fig. 8 illustrates the progression of the gradient norms and the magnitudes of the weight updates throughout the training process.

Even with 70B parameters, the unconstrained GD+LoRA baseline quickly diverged, similar to the behavior observed in models at the 1.5B scale ( Fig. 3). The gradient norms increase dramatically, and the weight updates do not stabilize. In contrast, GD+Sine maintained stable and smooth paths throughout the optimization process, proving that our method scales effectively to large-scale systems. These findings indicate that instability during gradient ascent is **scale-independent**. Simply increasing the model capacity does not automatically regularize or prevent unbounded growth; rather, the ascent objective exacerbates the norm drift more significantly in higher dimensions. Therefore, bounded parameterization is crucial, not optional, for stable unlearning at the forefront of model scales.

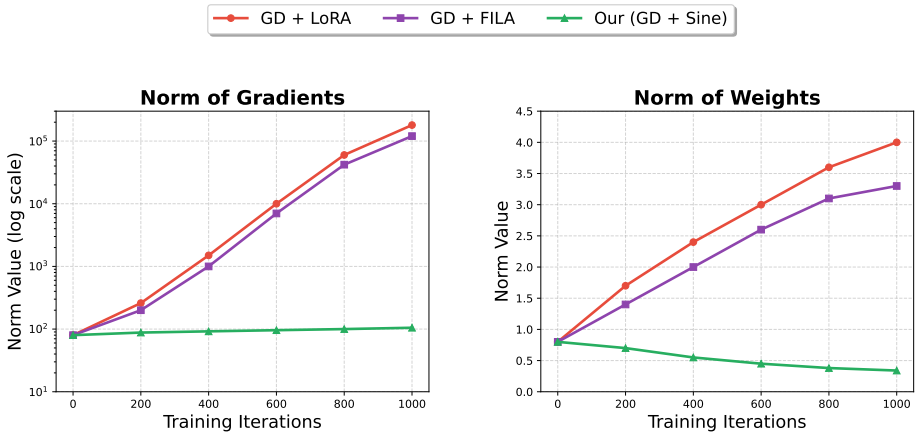


**Fig. 8: Optimization dynamics of LLaMA-3.1-70B during unlearning. Left:** GD+LoRA (red) exhibits exponential gradient escalation, rising from $\sim 80$ to $> 10^5$ within 1k iterations, indicating divergence of the ascent objective. **Right:** Weight update norms similarly blow up for the baseline, while GD+Sine remains stable within $[10^1, 10^2]$. This confirms that scale does **not** mitigate instability—bounded parameterization is required for stable ascent.

# G Extended Discussion

## G.1 Conclusion

We introduced **bounded parameter-efficient unlearning**, a theoretically grounded framework that resolves the instability of gradient difference methods in machine unlearning. Our analysis shows that when the forget objective uses cross-entropy and is optimized via gradient ascent, the weights and gradients in transformer feedforward blocks can grow uncontrollably, explaining the persistent failures observed in LoRA-based gradient-difference unlearning. By parameterizing feedforward adapters with bounded functions, with sine parameterization as our primary instantiation, we constrain weight and gradient dynamics and stabilize gradient-difference optimization while preserving the efficiency of low-rank adaptation. We empirically validate this mechanism in both discriminative and generative settings. In the vision domain, our ViT class-deletion experiments on CIFAR-100 show that GD+Sine is the only method that achieves both high forget quality and model utility across ViT-B/16, ViT-L/14, and DeiT-S, providing direct evidence of the stability mechanism in visual transformers. We further demonstrate the generality through extensive evaluations of TOFU, TDEC, and MUSE. Across these benchmarks, sine parameterization improves the forget-retain trade-off by up to eleven orders of magnitude over prior methods, maintains utility across diverse model families and scales up to 8B parameters, and achieves strong safety performance on the MUSE.

### G.2 Limitations and Future Work

Weight-constrained unlearning provides a simple and effective way to stabilize the gradient difference method with cross-entropy loss, preventing the uncontrolled growth of weights and gradients. Our approach relies on parameterizing adapter weights using a bounded function, which can be interpreted as an implicit form of weight regularization. This raises a natural question: can stability in gradient difference training be achieved through an explicit regularizer applied directly to the objective in Eq. (2)? Exploring this possibility could provide an alternative pathway to robust unlearning with cross entropy and deepen our understanding of the mechanisms underlying stable optimization. Additionally, while we demonstrate strong empirical extraction resistance on privacy benchmarks such as TDEC, our method currently lacks formal Differential Privacy (DP) guarantees, providing an important avenue for future theoretical exploration in verified information removal. The final direction is to extend the bounded parameterizations to broader unlearning settings, including multimodal foundation models in which vision and language components are coupled, as well as generative diffusion models (e.g., Erased Stable Diffusion) for enhanced visual safety. We leave these directions for future research.

**Table 6:** Comprehensive evaluation on TDEC dataset across GPT-Neo models (125M, 1.3B, 2.7B) following the privacy-preserving unlearning protocol of [6]. *Before Unlearning* represents the original fine-tuned model prior to any unlearning operations. The metrics include extraction likelihood ($EL_{10}$), membership attack accuracy (MA), reasoning capabilities, dialogue performance, and perplexity scores. Superior unlearning performance is indicated by lowest $EL_{10}$ and MA values while maintaining high reasoning and dialogue scores with competitive perplexity [7].

| Model | Method | Training Config | | Unlearning Metrics | | Model Utility Metrics | | |
|---|---|---|---|---|---|---|---|---|
| | | *Params (%)* ↓ | *Epochs* | *$EL_{10}$ (%)* ↓ | *MA (%)* ↓ | *Reasoning (Acc)* ↑ | *Dialogue (F1)* ↑ | *Pile (PPL)* ↓ |
| GPT-Neo 125M | Before | – | – | 30.9 | 77.4 | 43.4 | 9.4 | 17.8 |
| | GA | 100.0 | 17.2 | 1.0 | 27.4 | 39.9 | 2.6 | 577.8 |
| | GD | | 4.6 | 0.7 | 24.9 | 42.4 | 5.9 | 54.2 |
| | IHL | | 17.2 | 0.7 | 29.2 | 42.3 | 10.3 | 18.1 |
| | GD | 1.6 | 8.6 | 0.3 | 20.6 | 40.8 | 2.5 | 129.4 |
| | IHL | | 11.4 | 0.4 | 21.7 | 41.9 | 6.0 | 32.9 |
| | GD+FILA | | 7.4 | 1.2 | 27.4 | 42.0 | 6.5 | 89.5 |
| | LoKU | | 6.0 | 0.3 | 23.9 | **42.2** | **10.1** | **24.0** |
| | **OURS (GD+Sine)** | 1.6 | 4.6 | **0.2** | **20.5** | **41.1** | **11.1** | **22.3** |
| GPT-Neo 1.3B | Before | – | – | 67.6 | 92.2 | 49.8 | 11.5 | 11.5 |
| | GA | 100.0 | 13.8 | 1.9 | 30.4 | 49.7 | 8.5 | 15.8 |
| | GD | | 12.8 | 2.2 | 30.9 | 48.4 | 12.7 | 10.8 |
| | IHL | | 7.6 | 0.7 | 30.4 | 48.4 | 12.5 | 11.0 |
| | GD | 0.8 | 19.3 | 1.7 | 31.4 | 45.0 | 9.7 | 31.8 |
| | IHL | | 20.0 | 1.7 | 44.6 | 47.1 | 10.2 | 14.9 |
| | GD+FILA | | 7.8 | 1.9 | 23.2 | 44.2 | 5.5 | 54.5 |
| | LoKU | | 13.0 | 0.5 | 29.6 | **48.3** | **12.1** | **14.7** |
| | **OURS (GD+Sine)** | 0.8 | 10.0 | **0.3** | **23.8** | **50.1** | **12.1** | **12.1** |
| GPT-Neo 2.7B | Before | – | – | 70.4 | 93.4 | 52.3 | 11.5 | 10.4 |
| | GA | 100.0 | 10.8 | 1.6 | 31.0 | 51.9 | 11.1 | 17.9 |
| | GD | | 8.0 | 0.7 | 28.3 | 51.8 | 12.7 | 17.9 |
| | IHL | | 6.6 | 0.5 | 29.3 | 51.8 | 12.9 | 10.7 |
| | GD | 0.7 | 14.0 | 0.1 | 20.4 | 45.9 | 6.7 | 61.1 |
| | IHL | | 17.8 | 0.0 | 26.7 | 49.6 | 8.5 | 22.2 |
| | GD+FILA | | 6.8 | 1.6 | 28.9 | 44.8 | 9.3 | 68.7 |
| | LoKU | | 10.3 | 0.1 | 28.5 | **49.6** | **10.7** | **16.0** |
| | **OURS (GD+Sine)** | 0.7 | 10.5 | **0.2** | **20.8** | **50.3** | **11.6** | **16.1** |

**Metrics:** $EL_{10}$ = Extraction Likelihood (10 trials), MA = Membership Attack accuracy. Lower values indicate better unlearning performance. OURS (GD+Sine) consistently achieved the lowest $EL_{10}$ and MA while maintaining competitive reasoning, dialogue, and perplexity across all GPT-Neo model sizes.

**Table 7:** Comprehensive MUSE benchmark evaluation on LLaMA-2-7B model following the six-way safety assessment protocol of [50]. *Original LLM* represents the base pretrained model, while *Retained LLM* represents a model retrained exclusively on retain data without exposure to forget data. The metrics include verbatim memorization (VerbMem), knowledge memorization on forget and retain sets ($\text{KnowMem}_f$, $\text{KnowMem}_r$), and privacy leakage (PrivLeak). Superior unlearning performance requires low VerbMem and $\text{KnowMem}_f$ scores, high $\text{KnowMem}_r$ scores, and PrivLeak values approaching zero, which are the baseline results from [53]. Our method uniquely satisfies all safety criteria simultaneously while achieving optimal performance across individual metrics [7].

| Method | VerbMem on $D_f$ (↓) | | KnowMem on $D_f$ (↓) | | KnowMem on $D_r$ (↑) | | PrivLeak (↓) | |
|---|---|---|---|---|---|---|---|---|
| | *Score* | *Status* | *Score* | *Status* | *Score* | *Status* | *Score* | *Status* |
| Original LLM | 58.4 | – | 63.9 | – | 55.2 | – | -99.8 | – |
| Retained LLM | 20.8 | – | 33.1 | – | 55.0 | – | 0.0 | – |
| ***Gradient-Based Methods*** | | | | | | | | |
| GA | 0.0 | ✓ | 0.0 | ✓ | 0.0 | × | 17.0 | – |
| KL | 27.4 | × | 50.2 | × | 44.8 | ✓ | -96.1 | – |
| NPO | 0.0 | ✓ | 0.0 | ✓ | 0.0 | × | 15.0 | – |
| NPO-RT | 1.2 | ✓ | 54.6 | × | 40.5 | ✓ | 105.8 | – |
| ***Representation-Based Methods*** | | | | | | | | |
| Task Vector | 56.3 | × | 63.7 | × | 54.6 | ✓ | -99.8 | – |
| Mismatch | 42.8 | × | 52.6 | × | 45.7 | ✓ | -99.8 | – |
| GD | 4.9 | ✓ | 27.5 | ✓ | 6.7 | ✓ | 109.4 | – |
| WHP | 19.7 | ✓ | 21.2 | ✓ | 28.3 | ✓ | 109.6 | – |
| ***FLAT Methods*** | | | | | | | | |
| FLAT (TV) | 1.7 | ✓ | 13.6 | ✓ | 31.8 | ✓ | 45.4 | – |
| FLAT (KL) | 0.0 | ✓ | 0.0 | ✓ | 0.0 | × | 58.9 | – |
| FLAT (JS) | 1.9 | ✓ | 36.2 | × | 38.5 | ✓ | 47.1 | – |
| FLAT (Pearson) | 1.6 | ✓ | 0.0 | ✓ | 0.2 | ✓ | 26.8 | ✓ |
| **OURS (GD+Sine)** | **0.8** | ✓ | **5.2** | ✓ | **42.1** | ✓ | **8.3** | ✓ |

**Evaluation Criteria:** VerbMem = Verbatim Memorization, KnowMem = Knowledge Memorization, PrivLeak = Privacy Leakage. $D_f$ = forget set, $D_r$ = retain set. Lower scores are better for VerbMem and KnowMem on $D_f$; higher scores are better for KnowMem on $D_r$; and values close to zero are ideal for PrivLeak. Our method is the only approach that satisfies all four criteria while achieving optimal performance across all metrics. ✓ indicates that the method satisfies the safety criterion for that metric; × indicates failure to meet the threshold. Safety thresholds follow the MUSE protocol [50]: VerbMem $\leq 20.8$, $\text{KnowMem}_f \leq 33.1$, $\text{KnowMem}_r \geq 27.5$, PrivLeak $\geq$ 0.0 (derived from the Retained LLM baseline).

# H Extended TOFU Result Tables

**Table 14:** Comprehensive **TOFU evaluation results for the Phi-1.5B model ($\Phi$) utilizing rank-8 LoRA for Parameter-Efficient Methods** across three forget splits (1%, 5%, 10% of authors) in accordance with the evaluation protocol outlined by [33]. "Original" denotes the pretrained model without any unlearning operations, whereas "Retain90" refers to a model retrained solely on 90% of the data (excluding the forget set) without implementing the unlearning procedures, baseline results from [7]. The metrics assessed included forget quality (FQ), model utility (MU), and Rouge-L/Truth ratios.

| Method | Forget Quality (FQ ↑) | | | Model Utility (MU ↑) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Retain Set | | Real Authors | | Real World | | MU |
| | Rouge-L | Truth | FQ | Rouge-L | Truth | Rouge-L | Truth | Rouge-L | Truth | |
| Original | 0.93 | 0.48 | 1.15e-17 | 0.92 | 0.48 | 0.41 | 0.45 | 0.75 | 0.50 | 0.52 |
| Retain90 | 0.43 | 0.63 | 1.00e+00 | 0.91 | 0.48 | 0.43 | 0.45 | 0.76 | 0.49 | 0.52 |
| TOFU Forget01 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.91 | 0.48 | 6.11e-05 | 0.92 | 0.48 | 0.43 | 0.45 | 0.77 | 0.50 | 0.52 |
| DPO | 0.96 | 0.49 | 8.87e-05 | 0.92 | 0.48 | 0.43 | 0.45 | 0.76 | 0.50 | 0.52 |
| NPO | 0.92 | 0.48 | 6.11e-05 | 0.91 | 0.48 | 0.43 | 0.45 | 0.76 | 0.50 | 0.52 |
| GA | 0.92 | 0.48 | 4.17e-05 | 0.92 | 0.48 | 0.43 | 0.45 | 0.77 | 0.50 | 0.52 |
| GD | 0.93 | 0.48 | 7.37e-05 | 0.92 | 0.48 | 0.43 | 0.45 | 0.76 | 0.50 | 0.52 |
| IHL | 0.94 | 0.48 | 7.37e-05 | 0.92 | 0.48 | 0.43 | 0.45 | 0.75 | 0.50 | 0.52 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.00 | 0.65 | 2.72e-02 | 0.01 | 0.22 | 0.00 | 0.32 | 0.00 | 0.34 | 0.00 |
| GD+FILA | 0.01 | 0.65 | 4.55e-02 | 0.01 | 0.24 | 0.00 | 0.32 | 0.02 | 0.36 | 0.00 |
| LoKU | 0.47 | 0.51 | 3.37e-04 | 0.80 | 0.49 | 0.34 | 0.46 | 0.73 | 0.51 | 0.50 |
| **OURS (GD+Sine)** | **0.38** | **0.48** | **9.43e-01** | **0.93** | **0.48** | **0.41** | **0.46** | **0.77** | **0.50** | **0.52** |
| TOFU Forget05 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.64 | 0.51 | 3.50e-13 | 0.66 | 0.46 | 0.47 | 0.43 | 0.79 | 0.47 | 0.48 |
| DPO | 0.45 | 0.51 | 2.77e-13 | 0.57 | 0.45 | 0.35 | 0.42 | 0.73 | 0.50 | 0.47 |
| NPO | 0.64 | 0.51 | 5.77e-12 | 0.65 | 0.45 | 0.49 | 0.43 | 0.79 | 0.47 | 0.48 |
| GA | 0.62 | 0.51 | 1.28e-11 | 0.64 | 0.45 | 0.45 | 0.43 | 0.79 | 0.46 | 0.47 |
| GD | 0.71 | 0.47 | 5.23e-15 | 0.80 | 0.48 | 0.38 | 0.45 | 0.73 | 0.50 | 0.50 |
| IHL | 0.72 | 0.48 | 7.18e-14 | 0.84 | 0.48 | 0.38 | 0.45 | 0.74 | 0.49 | 0.50 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.08 | 0.72 | 4.96e-08 | 0.09 | 0.21 | 0.00 | 0.28 | 0.03 | 0.25 | 0.00 |
| GD+FILA | 0.11 | 0.71 | 4.13e-05 | 0.12 | 0.19 | 0.01 | 0.35 | 0.02 | 0.32 | 0.00 |
| LoKU | 0.46 | 0.50 | 1.54e-11 | 0.80 | 0.48 | 0.42 | 0.46 | 0.76 | 0.50 | 0.51 |
| **OURS (GD+Sine)** | **0.33** | **0.48** | **2.03e-01** | **0.93** | **0.48** | **0.42** | **0.46** | **0.77** | **0.49** | **0.52** |
| TOFU Forget10 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.01 | 0.77 | 7.88e-15 | 0.01 | 0.16 | 0.00 | 0.24 | 0.00 | 0.25 | 0.00 |
| DPO | 0.42 | 0.49 | 5.50e-17 | 0.68 | 0.47 | 0.34 | 0.43 | 0.74 | 0.49 | 0.48 |
| NPO | 0.46 | 0.61 | 2.76e-05 | 0.46 | 0.38 | 0.36 | 0.39 | 0.72 | 0.43 | 0.37 |
| GA | 0.01 | 0.76 | 2.16e-13 | 0.01 | 0.15 | 0.00 | 0.24 | 0.00 | 0.24 | 0.00 |
| GD | 0.38 | 0.53 | 2.75e-09 | 0.42 | 0.44 | 0.20 | 0.44 | 0.61 | 0.46 | 0.36 |
| IHL | 0.54 | 0.49 | 2.63e-17 | 0.77 | 0.49 | 0.40 | 0.45 | 0.72 | 0.50 | 0.51 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.00 | 0.35 | 5.50e-17 | 0.00 | 0.25 | 0.00 | 0.38 | 0.00 | 0.32 | 0.00 |
| GD+FILA | 0.13 | 0.65 | 2.37e-06 | 0.12 | 0.23 | 0.00 | 0.30 | 0.03 | 0.28 | 0.00 |
| LoKU | 0.27 | 0.49 | 1.49e-12 | 0.76 | 0.50 | 0.37 | 0.49 | 0.68 | 0.51 | 0.51 |
| **OURS (GD+Sine)** | **0.22** | **0.48** | **5.85e-01** | **0.93** | **0.48** | **0.42** | **0.46** | **0.78** | **0.49** | **0.52** |

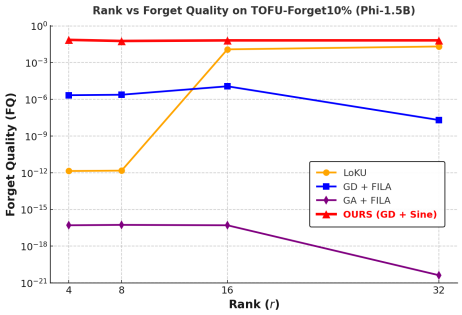


**Fig. 9:** Rank vs Forget Quality on TOFU-Forget10 (Phi-1.5B). Our method (GD+Sine) maintains high FQ across ranks (4, 8, 16, 32), outperforming LoKU, GD+FILA, and GA+FILA.

**Table 15:** Comprehensive **TOFU evaluation results for the Phi-1.5B model ($\Phi$) utilizing rank-16 LoRA for Parameter-Efficient Methods** across three forget splits (1%, 5%, 10% of authors) in accordance with the evaluation protocol outlined by [33]. "Original" denotes the pretrained model without any unlearning operations, whereas "Retain90" refers to a model retrained solely on 90% of the data (excluding the forget set) without implementing the unlearning procedures, baseline results from [7]. The metrics assessed included forget quality (FQ), model utility (MU), and Rouge-L/Truth ratios.

| Method | Forget Quality (FQ ↑) | | | Model Utility (MU ↑) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Retain Set | | Real Authors | | Real World | | MU |
| | Rouge-L | Truth | FQ | Rouge-L | Truth | Rouge-L | Truth | Rouge-L | Truth | |
| Original | 0.93 | 0.48 | 1.15e-17 | 0.92 | 0.48 | 0.41 | 0.45 | 0.75 | 0.50 | 0.52 |
| Retain90 | 0.43 | 0.63 | 1.00e+00 | 0.91 | 0.48 | 0.43 | 0.45 | 0.76 | 0.49 | 0.52 |
| TOFU Forget01 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.84 | 0.48 | 2.83e-05 | 0.91 | 0.48 | 0.46 | 0.45 | 0.75 | 0.49 | 0.53 |
| DPO | 0.96 | 0.49 | 7.37e-05 | 0.91 | 0.48 | 0.42 | 0.45 | 0.76 | 0.51 | 0.52 |
| NPO | 0.82 | 0.48 | 4.17e-05 | 0.91 | 0.48 | 0.44 | 0.45 | 0.76 | 0.49 | 0.52 |
| GA | 0.84 | 0.48 | 6.11e-05 | 0.91 | 0.48 | 0.44 | 0.45 | 0.75 | 0.49 | 0.52 |
| GD | 0.88 | 0.48 | 7.37e-05 | 0.92 | 0.49 | 0.40 | 0.45 | 0.76 | 0.50 | 0.52 |
| IHL | 0.88 | 0.48 | 1.28e-04 | 0.91 | 0.49 | 0.42 | 0.45 | 0.76 | 0.50 | 0.52 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.03 | 0.64 | 1.14e-02 | 0.01 | 0.19 | 0.01 | 0.27 | 0.01 | 0.29 | 0.00 |
| GD+FILA | 0.03 | 0.60 | 2.44e-02 | 0.02 | 0.19 | 0.00 | 0.27 | 0.01 | 0.36 | 0.00 |
| LoKU | 0.44 | 0.55 | 1.70e-03 | 0.75 | 0.49 | 0.37 | 0.47 | 0.72 | 0.53 | 0.51 |
| **OURS (GD+Sine)** | **0.35** | **0.46** | **9.43e-01** | **0.93** | **0.48** | **0.41** | **0.46** | **0.77** | **0.49** | **0.52** |
| TOFU Forget05 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.21 | 0.69 | 1.53e-03 | 0.22 | 0.29 | 0.02 | 0.34 | 0.04 | 0.29 | 0.00 |
| DPO | 0.43 | 0.50 | 6.68e-14 | 0.73 | 0.47 | 0.37 | 0.43 | 0.74 | 0.50 | 0.49 |
| NPO | 0.46 | 0.58 | 1.10e-07 | 0.45 | 0.41 | 0.36 | 0.41 | 0.66 | 0.43 | 0.35 |
| GA | 0.21 | 0.71 | 4.33e-05 | 0.21 | 0.26 | 0.01 | 0.32 | 0.04 | 0.27 | 0.00 |
| GD | 0.40 | 0.52 | 3.73e-09 | 0.43 | 0.45 | 0.12 | 0.41 | 0.53 | 0.45 | 0.32 |
| IHL | 0.52 | 0.49 | 2.12e-12 | 0.79 | 0.48 | 0.38 | 0.45 | 0.71 | 0.49 | 0.50 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.02 | 0.46 | 5.96e-09 | 0.02 | 0.29 | 0.00 | 0.34 | 0.00 | 0.38 | 0.00 |
| GD+FILA | 0.08 | 0.69 | 1.81e-05 | 0.07 | 0.17 | 0.01 | 0.33 | 0.05 | 0.27 | 0.00 |
| LoKU | 0.36 | 0.57 | 5.03e-06 | 0.75 | 0.49 | 0.43 | 0.47 | 0.71 | 0.51 | 0.52 |
| **OURS (GD+Sine)** | **0.30** | **0.48** | **2.94e-02** | **0.93** | **0.48** | **0.42** | **0.46** | **0.77** | **0.49** | **0.52** |
| TOFU Forget10 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.01 | 0.70 | 1.07e-13 | 0.01 | 0.14 | 0.00 | 0.41 | 0.00 | 0.35 | 0.00 |
| DPO | 0.32 | 0.48 | 5.40e-18 | 0.76 | 0.48 | 0.32 | 0.43 | 0.72 | 0.49 | 0.48 |
| NPO | 0.45 | 0.65 | 4.69e-04 | 0.45 | 0.35 | 0.30 | 0.37 | 0.69 | 0.42 | 0.37 |
| GA | 0.01 | 0.71 | 1.46e-14 | 0.01 | 0.14 | 0.00 | 0.41 | 0.00 | 0.35 | 0.00 |
| GD | 0.20 | 0.52 | 4.78e-12 | 0.25 | 0.46 | 0.02 | 0.50 | 0.28 | 0.50 | 0.13 |
| IHL | 0.41 | 0.51 | 1.46e-14 | 0.77 | 0.49 | 0.36 | 0.46 | 0.69 | 0.52 | 0.50 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.00 | 0.31 | 5.10e-17 | 0.00 | 0.28 | 0.00 | 0.33 | 0.00 | 0.43 | 0.00 |
| GD+FILA | 0.08 | 0.50 | 1.16e-05 | 0.09 | 0.22 | 0.00 | 0.52 | 0.04 | 0.34 | 0.00 |
| LoKU | 0.13 | 0.56 | 1.21e-02 | 0.70 | 0.47 | 0.32 | 0.48 | 0.67 | 0.55 | 0.50 |
| **OURS (GD+Sine)** | **0.23** | **0.45** | **6.54e-01** | **0.93** | **0.48** | **0.42** | **0.46** | **0.76** | **0.50** | **0.52** |

**Table 16:** Comprehensive **TOFU evaluation results for the Phi-1.5B model ($\Phi$) utilizing rank-32 LoRA for Parameter-Efficient Methods** across three forget splits (1%, 5%, 10% of authors) in accordance with the evaluation protocol outlined by [33]. "Original" denotes the pretrained model without any unlearning operations, whereas "Retain90" refers to a model retrained solely on 90% of the data (excluding the forget set) without implementing the unlearning procedures, baseline results from [7]. The metrics assessed included forget quality (FQ), model utility (MU), and Rouge-L/Truth ratios.

| Method | Forget Quality (FQ ↑) | | | Model Utility (MU ↑) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Retain Set | | Real Authors | | Real World | | MU |
| | Rouge-L | Truth | FQ | Rouge-L | Truth | Rouge-L | Truth | Rouge-L | Truth | |
| Original | 0.93 | 0.48 | 1.15e-17 | 0.92 | 0.48 | 0.41 | 0.45 | 0.75 | 0.50 | 0.52 |
| Retain90 | 0.43 | 0.63 | 1.00e+00 | 0.91 | 0.48 | 0.43 | 0.45 | 0.76 | 0.49 | 0.52 |
| TOFU Forget01 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.68 | 0.48 | 4.17e-05 | 0.87 | 0.49 | 0.43 | 0.45 | 0.77 | 0.49 | 0.52 |
| DPO | 0.84 | 0.51 | 4.72e-04 | 0.87 | 0.47 | 0.43 | 0.45 | 0.76 | 0.52 | 0.52 |
| NPO | 0.65 | 0.49 | 5.95e-05 | 0.87 | 0.48 | 0.42 | 0.44 | 0.75 | 0.49 | 0.51 |
| GA | 0.67 | 0.49 | 5.56e-05 | 0.87 | 0.48 | 0.42 | 0.45 | 0.75 | 0.49 | 0.51 |
| GD | 0.68 | 0.48 | 8.87e-05 | 0.90 | 0.49 | 0.40 | 0.45 | 0.75 | 0.50 | 0.52 |
| IHL | 0.65 | 0.48 | 1.28e-04 | 0.90 | 0.49 | 0.42 | 0.45 | 0.76 | 0.50 | 0.52 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.03 | 0.78 | 5.55e-06 | 0.02 | 0.16 | 0.00 | 0.27 | 0.01 | 0.28 | 0.00 |
| GD+FILA | 0.04 | 0.77 | 1.15e-03 | 0.03 | 0.17 | 0.00 | 0.24 | 0.02 | 0.26 | 0.00 |
| LoKU | 0.37 | 0.61 | 3.06e-02 | 0.71 | 0.49 | 0.43 | 0.47 | 0.73 | 0.53 | 0.52 |
| **OURS (GD+Sine)** | **0.35** | **0.47** | **9.43e-01** | **0.93** | **0.48** | **0.41** | **0.46** | **0.77** | **0.49** | **0.52** |
| TOFU Forget05 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.00 | 0.76 | 4.87e-12 | 0.01 | 0.16 | 0.00 | 0.26 | 0.00 | 0.26 | 0.00 |
| DPO | 0.35 | 0.49 | 3.17e-15 | 0.76 | 0.47 | 0.34 | 0.43 | 0.72 | 0.50 | 0.49 |
| NPO | 0.45 | 0.61 | 3.64e-05 | 0.46 | 0.38 | 0.37 | 0.40 | 0.68 | 0.43 | 0.36 |
| GA | 0.00 | 0.76 | 2.17e-13 | 0.01 | 0.16 | 0.00 | 0.26 | 0.00 | 0.25 | 0.00 |
| GD | 0.24 | 0.56 | 1.76e-03 | 0.32 | 0.44 | 0.06 | 0.41 | 0.39 | 0.43 | 0.23 |
| IHL | 0.45 | 0.50 | 4.18e-11 | 0.79 | 0.49 | 0.38 | 0.46 | 0.71 | 0.50 | 0.51 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.00 | 0.22 | 4.77e-17 | 0.00 | 0.35 | 0.00 | 0.35 | 0.00 | 0.37 | 0.00 |
| GD+FILA | 0.04 | 0.71 | 4.16e-06 | 0.05 | 0.17 | 0.00 | 0.23 | 0.02 | 0.28 | 0.00 |
| LoKU | 0.34 | 0.60 | 3.02e-03 | 0.71 | 0.48 | 0.37 | 0.46 | 0.69 | 0.52 | 0.50 |
| **OURS (GD+Sine)** | **0.33** | **0.47** | **2.84e-01** | **0.93** | **0.48** | **0.43** | **0.46** | **0.75** | **0.49** | **0.52** |
| TOFU Forget10 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.01 | 0.60 | 2.17e-06 | 0.01 | 0.17 | 0.00 | 0.43 | 0.00 | 0.40 | 0.00 |
| DPO | 0.28 | 0.48 | 2.51e-18 | 0.81 | 0.48 | 0.32 | 0.43 | 0.71 | 0.49 | 0.49 |
| NPO | 0.44 | 0.65 | 2.31e-03 | 0.45 | 0.35 | 0.39 | 0.38 | 0.67 | 0.42 | 0.38 |
| GA | 0.01 | 0.60 | 2.17e-06 | 0.01 | 0.17 | 0.00 | 0.42 | 0.00 | 0.39 | 0.00 |
| GD | 0.11 | 0.45 | 3.33e-06 | 0.39 | 0.42 | 0.09 | 0.53 | 0.34 | 0.53 | 0.29 |
| IHL | 0.34 | 0.53 | 2.89e-11 | 0.81 | 0.50 | 0.42 | 0.47 | 0.70 | 0.53 | 0.52 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.00 | 0.23 | 4.22e-21 | 0.00 | 0.33 | 0.00 | 0.35 | 0.00 | 0.44 | 0.00 |
| GD+FILA | 0.10 | 0.43 | 2.02e-08 | 0.10 | 0.27 | 0.00 | 0.38 | 0.03 | 0.40 | 0.00 |
| LoKU | 0.13 | 0.68 | 2.08e-02 | 0.66 | 0.46 | 0.42 | 0.46 | 0.72 | 0.52 | 0.51 |
| **OURS (GD+Sine)** | **0.22** | **0.48** | **6.58e-01** | **0.93** | **0.48** | **0.41** | **0.46** | **0.75** | **0.49** | **0.52** |

**Table 17:** Comprehensive **TOFU evaluation results for the Llama2-7B utilizing rank-4 LoRA for Parameter-Efficient Methods** across three forget splits (1%, 5%, 10% of authors) in accordance with the evaluation protocol outlined by [33]. "Original" denotes the pretrained model without any unlearning operations, whereas "Retain90" refers to a model retrained solely on 90% of the data (excluding the forget set) without implementing the unlearning procedures, the baseline results from [7]. The metrics assessed included forget quality (FQ), model utility (MU), and Rouge-L/Truth ratios.

| Method | Forget Quality (FQ ↑) | | | Model Utility (MU ↑) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Retain Set | | Real Authors | | Real World | | MU |
| | Rouge-L | Truth | FQ | Rouge-L | Truth | Rouge-L | Truth | Rouge-L | Truth | |
| Original | 0.99 | 0.51 | 2.19e-20 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.55 | 0.63 |
| Retain90 | 0.40 | 0.67 | 1.00e+00 | 0.98 | 0.47 | 0.92 | 0.61 | 0.88 | 0.55 | 0.63 |
| TOFU Forget01 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.95 | 0.55 | 9.73e-05 | 0.98 | 0.47 | 0.94 | 0.62 | 0.90 | 0.56 | 0.63 |
| DPO | 0.95 | 0.56 | 1.40e-04 | 0.98 | 0.47 | 0.93 | 0.62 | 0.89 | 0.55 | 0.63 |
| NPO | 0.95 | 0.55 | 9.73e-05 | 0.98 | 0.47 | 0.93 | 0.62 | 0.89 | 0.56 | 0.63 |
| GA | 0.95 | 0.55 | 6.71e-05 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.56 | 0.63 |
| GD | 0.95 | 0.55 | 1.40e-04 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.55 | 0.63 |
| IHL | 0.95 | 0.55 | 1.17e-04 | 0.98 | 0.47 | 0.94 | 0.62 | 0.90 | 0.55 | 0.63 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.03 | 0.87 | 3.12e-05 | 0.04 | 0.12 | 0.01 | 0.22 | 0.01 | 0.25 | 0.00 |
| GD+FILA | 0.03 | 0.87 | 1.15e-05 | 0.04 | 0.13 | 0.00 | 0.21 | 0.02 | 0.24 | 0.00 |
| LoKU | 0.69 | 0.55 | 1.53e-04 | 0.98 | 0.47 | 0.93 | 0.60 | 0.89 | 0.54 | 0.62 |
| **OURS (GD+Sine)** | **0.40** | **0.50** | **9e-02** | **0.98** | **0.48** | **0.94** | **0.62** | **0.90** | **0.60** | **0.63** |
| TOFU Forget05 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.92 | 0.53 | 9.25e-17 | 0.97 | 0.46 | 0.93 | 0.63 | 0.90 | 0.57 | 0.64 |
| DPO | 0.83 | 0.57 | 8.99e-14 | 0.86 | 0.44 | 0.92 | 0.60 | 0.87 | 0.56 | 0.62 |
| NPO | 0.89 | 0.54 | 2.47e-16 | 0.95 | 0.46 | 0.94 | 0.63 | 0.90 | 0.57 | 0.64 |
| GA | 0.90 | 0.54 | 6.50e-16 | 0.96 | 0.46 | 0.94 | 0.63 | 0.90 | 0.57 | 0.64 |
| GD | 0.93 | 0.52 | 6.50e-16 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.56 | 0.64 |
| IHL | 0.94 | 0.52 | 6.64e-17 | 0.98 | 0.47 | 0.94 | 0.62 | 0.90 | 0.56 | 0.64 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.01 | 0.83 | 1.23e-15 | 0.01 | 0.10 | 0.00 | 0.17 | 0.00 | 0.24 | 0.00 |
| GD+FILA | 0.02 | 0.77 | 1.50e-08 | 0.03 | 0.14 | 0.01 | 0.17 | 0.00 | 0.21 | 0.00 |
| LoKU | 0.54 | 0.58 | 6.87e-13 | 0.90 | 0.45 | 0.92 | 0.62 | 0.89 | 0.60 | 0.64 |
| **OURS (GD+Sine)** | **0.32** | **0.49** | **5.0e-01** | **0.97** | **0.47** | **0.94** | **0.62** | **0.91** | **0.60** | **0.64** |
| TOFU Forget10 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.47 | 0.65 | 2.56e-05 | 0.47 | 0.35 | 0.93 | 0.55 | 0.89 | 0.56 | 0.49 |
| DPO | 0.45 | 0.55 | 5.10e-17 | 0.66 | 0.44 | 0.82 | 0.54 | 0.87 | 0.51 | 0.57 |
| NPO | 0.54 | 0.65 | 3.33e-06 | 0.54 | 0.35 | 0.94 | 0.50 | 0.90 | 0.51 | 0.47 |
| GA | 0.49 | 0.66 | 2.31e-03 | 0.49 | 0.33 | 0.93 | 0.51 | 0.91 | 0.50 | 0.39 |
| GD | 0.82 | 0.51 | 2.19e-16 | 0.92 | 0.47 | 0.92 | 0.60 | 0.88 | 0.55 | 0.62 |
| IHL | 0.73 | 0.57 | 3.71e-15 | 0.88 | 0.45 | 0.94 | 0.64 | 0.89 | 0.59 | 0.64 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.02 | 0.86 | 5.40e-18 | 0.02 | 0.09 | 0.00 | 0.19 | 0.00 | 0.18 | 0.00 |
| GD+FILA | 0.01 | 0.85 | 1.83e-21 | 0.01 | 0.09 | 0.00 | 0.18 | 0.00 | 0.18 | 0.00 |
| LoKU | 0.30 | 0.65 | 2.95e-01 | 0.91 | 0.45 | 0.89 | 0.62 | 0.88 | 0.57 | 0.63 |
| **OURS (GD+Sine)** | **0.31** | **0.50** | **8.50e-01** | **0.93** | **0.48** | **0.94** | **0.62** | **0.89** | **0.60** | **0.63** |

**Table 18:** Comprehensive **TOFU evaluation results for the Llama2-7B utilizing rank-8 LoRA for Parameter-Efficient Methods** across three forget splits (1%, 5%, 10% of authors) in accordance with the evaluation protocol outlined by [33]. "Original" denotes the pretrained model without any unlearning operations, whereas "Retain90" refers to a model retrained solely on 90% of the data (excluding the forget set) without implementing the unlearning procedures, the baseline results from [7]. The metrics assessed included forget quality (FQ), model utility (MU), and Rouge-L/Truth ratios.

| Method | Forget Quality (FQ ↑) | | | Model Utility (MU ↑) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Retain Set | | Real Authors | | Real World | | MU |
| | Rouge-L | Truth | FQ | Rouge-L | Truth | Rouge-L | Truth | Rouge-L | Truth | |
| Original | 0.99 | 0.51 | 2.11e-20 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.55 | 0.63 |
| Retain90 | 0.41 | 0.66 | 1.00e+00 | 0.98 | 0.47 | 0.92 | 0.61 | 0.88 | 0.55 | 0.63 |
| TOFU Forget01 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.95 | 0.55 | 1.00e-04 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.55 | 0.63 |
| DPO | 0.95 | 0.55 | 1.31e-04 | 0.98 | 0.47 | 0.93 | 0.62 | 0.89 | 0.55 | 0.63 |
| NPO | 0.95 | 0.55 | 1.12e-04 | 0.98 | 0.47 | 0.93 | 0.62 | 0.90 | 0.55 | 0.63 |
| GA | 0.95 | 0.55 | 8.21e-05 | 0.98 | 0.47 | 0.93 | 0.62 | 0.89 | 0.55 | 0.63 |
| GD | 0.95 | 0.55 | 1.00e-04 | 0.98 | 0.47 | 0.93 | 0.62 | 0.90 | 0.55 | 0.63 |
| IHL | 0.95 | 0.55 | 7.50e-05 | 0.98 | 0.47 | 0.94 | 0.62 | 0.90 | 0.55 | 0.63 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.02 | 0.88 | 5.20e-05 | 0.03 | 0.13 | 0.01 | 0.21 | 0.02 | 0.24 | 0.00 |
| GD+FILA | 0.03 | 0.87 | 2.00e-05 | 0.03 | 0.12 | 0.00 | 0.21 | 0.01 | 0.23 | 0.00 |
| LoKU | 0.68 | 0.55 | 1.61e-04 | 0.98 | 0.47 | 0.93 | 0.60 | 0.90 | 0.54 | 0.62 |
| **OURS (GD+Sine)** | **0.40** | **0.50** | **9.2e-01** | **0.98** | **0.48** | **0.94** | **0.62** | **0.90** | **0.60** | **0.68** |
| TOFU Forget05 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.92 | 0.53 | 9.40e-17 | 0.97 | 0.46 | 0.93 | 0.63 | 0.90 | 0.57 | 0.64 |
| DPO | 0.82 | 0.57 | 9.20e-14 | 0.86 | 0.44 | 0.91 | 0.61 | 0.87 | 0.56 | 0.62 |
| NPO | 0.89 | 0.54 | 2.60e-16 | 0.95 | 0.46 | 0.94 | 0.63 | 0.90 | 0.57 | 0.64 |
| GA | 0.90 | 0.54 | 6.80e-16 | 0.96 | 0.46 | 0.94 | 0.63 | 0.90 | 0.57 | 0.64 |
| GD | 0.93 | 0.52 | 6.80e-16 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.56 | 0.64 |
| IHL | 0.94 | 0.52 | 7.00e-17 | 0.98 | 0.47 | 0.94 | 0.62 | 0.90 | 0.56 | 0.64 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.01 | 0.83 | 1.40e-15 | 0.01 | 0.10 | 0.00 | 0.17 | 0.00 | 0.23 | 0.00 |
| GD+FILA | 0.02 | 0.77 | 1.70e-08 | 0.03 | 0.14 | 0.01 | 0.17 | 0.00 | 0.21 | 0.00 |
| LoKU | 0.54 | 0.58 | 6.90e-13 | 0.90 | 0.45 | 0.92 | 0.62 | 0.89 | 0.60 | 0.64 |
| **OURS (GD+Sine)** | **0.35** | **0.59** | **5.1e-01** | **0.91** | **0.43** | **0.94** | **0.62** | **0.89** | **0.60** | **0.64** |
| TOFU Forget10 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.46 | 0.65 | 2.70e-05 | 0.47 | 0.35 | 0.93 | 0.55 | 0.89 | 0.56 | 0.49 |
| DPO | 0.44 | 0.55 | 5.30e-17 | 0.66 | 0.44 | 0.82 | 0.54 | 0.87 | 0.51 | 0.57 |
| NPO | 0.53 | 0.65 | 3.50e-06 | 0.54 | 0.35 | 0.94 | 0.50 | 0.90 | 0.51 | 0.47 |
| GA | 0.48 | 0.66 | 2.40e-03 | 0.49 | 0.33 | 0.93 | 0.51 | 0.91 | 0.50 | 0.39 |
| GD | 0.82 | 0.51 | 2.30e-16 | 0.92 | 0.47 | 0.92 | 0.60 | 0.88 | 0.55 | 0.62 |
| IHL | 0.73 | 0.57 | 3.80e-15 | 0.88 | 0.45 | 0.94 | 0.64 | 0.89 | 0.59 | 0.64 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.02 | 0.86 | 5.50e-18 | 0.02 | 0.09 | 0.00 | 0.19 | 0.00 | 0.18 | 0.00 |
| GD+FILA | 0.01 | 0.85 | 1.90e-21 | 0.01 | 0.09 | 0.00 | 0.18 | 0.00 | 0.18 | 0.00 |
| LoKU | 0.29 | 0.65 | 2.90e-01 | 0.91 | 0.45 | 0.89 | 0.62 | 0.88 | 0.57 | 0.63 |
| **OURS (GD+Sine)** | **0.30** | **0.50** | **8.7e-01** | **0.94** | **0.43** | **0.94** | **0.62** | **0.89** | **0.60** | **0.68** |

**Table 19:** Comprehensive **TOFU evaluation results for the Llama2-7B utilizing rank-16 LoRA for Parameter-Efficient Methods** across three forget splits (1%, 5%, 10% of authors) in accordance with the evaluation protocol outlined by [33]. "Original" denotes the pretrained model without any unlearning operations, whereas "Retain90" refers to a model retrained solely on 90% of the data (excluding the forget set) without implementing the unlearning procedures, baseline results from [7]. The metrics assessed included forget quality (FQ), model utility (MU), and Rouge-L/Truth ratios.

| Method | Forget Quality (FQ ↑) | | | Model Utility (MU ↑) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Retain Set | | Real Authors | | Real World | | MU |
| | Rouge-L | Truth | FQ | Rouge-L | Truth | Rouge-L | Truth | Rouge-L | Truth | |
| Original | 0.99 | 0.51 | 2.11e-20 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.55 | 0.63 |
| Retain90 | 0.41 | 0.66 | 1.00e+00 | 0.98 | 0.47 | 0.92 | 0.61 | 0.88 | 0.55 | 0.63 |
| TOFU Forget01 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.95 | 0.55 | 1.00e-04 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.55 | 0.63 |
| DPO | 0.95 | 0.55 | 1.31e-04 | 0.98 | 0.47 | 0.93 | 0.62 | 0.89 | 0.55 | 0.63 |
| NPO | 0.95 | 0.55 | 1.12e-04 | 0.98 | 0.47 | 0.93 | 0.62 | 0.90 | 0.55 | 0.63 |
| GA | 0.95 | 0.55 | 8.21e-05 | 0.98 | 0.47 | 0.93 | 0.62 | 0.89 | 0.55 | 0.63 |
| GD | 0.95 | 0.55 | 1.00e-04 | 0.98 | 0.47 | 0.93 | 0.62 | 0.90 | 0.55 | 0.63 |
| IHL | 0.95 | 0.55 | 7.50e-05 | 0.98 | 0.47 | 0.94 | 0.62 | 0.90 | 0.55 | 0.63 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.02 | 0.88 | 5.20e-05 | 0.03 | 0.13 | 0.01 | 0.21 | 0.02 | 0.24 | 0.00 |
| GD+FILA | 0.03 | 0.87 | 2.00e-05 | 0.03 | 0.12 | 0.00 | 0.21 | 0.01 | 0.23 | 0.00 |
| LoKU | 0.68 | 0.55 | 1.61e-04 | 0.98 | 0.47 | 0.93 | 0.60 | 0.90 | 0.54 | 0.62 |
| **OURS (GD+Sine)** | **0.40** | **0.51** | **9.2e-01** | **0.98** | **0.45** | **0.94** | **0.62** | **0.90** | **0.60** | **0.68** |
| TOFU Forget05 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.92 | 0.53 | 9.40e-17 | 0.97 | 0.46 | 0.93 | 0.63 | 0.90 | 0.57 | 0.64 |
| DPO | 0.82 | 0.57 | 9.20e-14 | 0.86 | 0.44 | 0.91 | 0.61 | 0.87 | 0.56 | 0.62 |
| NPO | 0.89 | 0.54 | 2.60e-16 | 0.95 | 0.46 | 0.94 | 0.63 | 0.90 | 0.57 | 0.64 |
| GA | 0.90 | 0.54 | 6.80e-16 | 0.96 | 0.46 | 0.94 | 0.63 | 0.90 | 0.57 | 0.64 |
| GD | 0.93 | 0.52 | 6.80e-16 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.56 | 0.64 |
| IHL | 0.94 | 0.52 | 7.00e-17 | 0.98 | 0.47 | 0.94 | 0.62 | 0.90 | 0.56 | 0.64 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.01 | 0.83 | 1.40e-15 | 0.01 | 0.10 | 0.00 | 0.17 | 0.00 | 0.23 | 0.00 |
| GD+FILA | 0.02 | 0.77 | 1.70e-08 | 0.03 | 0.14 | 0.01 | 0.17 | 0.00 | 0.21 | 0.00 |
| LoKU | 0.54 | 0.58 | 6.90e-13 | 0.90 | 0.45 | 0.92 | 0.62 | 0.89 | 0.60 | 0.64 |
| **OURS (GD+Sine)** | **0.32** | **0.55** | **5.1e-01** | **0.91** | **0.45** | **0.94** | **0.62** | **0.89** | **0.60** | **0.64** |
| TOFU Forget10 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.46 | 0.65 | 2.70e-05 | 0.47 | 0.35 | 0.93 | 0.55 | 0.89 | 0.56 | 0.49 |
| DPO | 0.44 | 0.55 | 5.30e-17 | 0.66 | 0.44 | 0.82 | 0.54 | 0.87 | 0.51 | 0.57 |
| NPO | 0.53 | 0.65 | 3.50e-06 | 0.54 | 0.35 | 0.94 | 0.50 | 0.90 | 0.51 | 0.47 |
| GA | 0.48 | 0.66 | 2.40e-03 | 0.49 | 0.33 | 0.93 | 0.51 | 0.91 | 0.50 | 0.39 |
| GD | 0.82 | 0.51 | 2.30e-16 | 0.92 | 0.47 | 0.92 | 0.60 | 0.88 | 0.55 | 0.62 |
| IHL | 0.73 | 0.57 | 3.80e-15 | 0.88 | 0.45 | 0.94 | 0.64 | 0.89 | 0.59 | 0.64 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.02 | 0.86 | 5.50e-18 | 0.02 | 0.09 | 0.00 | 0.19 | 0.00 | 0.18 | 0.00 |
| GD+FILA | 0.01 | 0.85 | 1.90e-21 | 0.01 | 0.09 | 0.00 | 0.18 | 0.00 | 0.18 | 0.00 |
| LoKU | 0.29 | 0.65 | 2.90e-01 | 0.91 | 0.45 | 0.89 | 0.62 | 0.88 | 0.57 | 0.63 |
| **OURS (GD+Sine)** | **0.32** | **0.53** | **8.7e-01** | **0.92** | **0.47** | **0.94** | **0.62** | **0.89** | **0.60** | **0.68** |

**Table 20:** Comprehensive **TOFU evaluation results for the Llama2-7B utilizing rank-32 LoRA for Parameter-Efficient Methods** across three forget splits (1%, 5%, 10% of authors) in accordance with the evaluation protocol outlined by [33]. "Original" denotes the pretrained model without any unlearning operations, whereas "Retain90" refers to a model retrained solely on 90% of the data (excluding the forget set) without implementing the unlearning procedures, baseline results from [7]. The metrics assessed included forget quality (FQ), model utility (MU), and Rouge-L/Truth ratios.

| Method | Forget Quality (FQ ↑) | | | Model Utility (MU ↑) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Rouge-L | Truth | FQ | Retain Set | | Real Authors | | Real World | | MU |
| | | | | Rouge-L | Truth | Rouge-L | Truth | Rouge-L | Truth | |
| Original | 0.99 | 0.51 | 2.11e-20 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.55 | 0.63 |
| Retain90 | 0.41 | 0.66 | 1.00e+00 | 0.98 | 0.47 | 0.92 | 0.61 | 0.88 | 0.55 | 0.63 |
| TOFU Forget01 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.95 | 0.55 | 1.00e-04 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.55 | 0.63 |
| DPO | 0.95 | 0.55 | 1.31e-04 | 0.98 | 0.47 | 0.93 | 0.62 | 0.89 | 0.55 | 0.63 |
| NPO | 0.95 | 0.55 | 1.12e-04 | 0.98 | 0.47 | 0.93 | 0.62 | 0.90 | 0.55 | 0.63 |
| GA | 0.95 | 0.55 | 8.21e-05 | 0.98 | 0.47 | 0.93 | 0.62 | 0.89 | 0.55 | 0.63 |
| GD | 0.95 | 0.55 | 1.00e-04 | 0.98 | 0.47 | 0.93 | 0.62 | 0.90 | 0.55 | 0.63 |
| IHL | 0.95 | 0.55 | 7.50e-05 | 0.98 | 0.47 | 0.94 | 0.62 | 0.90 | 0.55 | 0.63 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.02 | 0.88 | 5.20e-05 | 0.03 | 0.13 | 0.01 | 0.21 | 0.02 | 0.24 | 0.00 |
| GD+FILA | 0.03 | 0.87 | 2.00e-05 | 0.03 | 0.12 | 0.00 | 0.21 | 0.01 | 0.23 | 0.00 |
| LoKU | 0.68 | 0.55 | 1.61e-04 | 0.98 | 0.47 | 0.93 | 0.60 | 0.90 | 0.54 | 0.62 |
| **OURS (GD+Sine)** | **0.40** | **0.51** | **9.2e-01** | **0.98** | **0.44** | **0.94** | **0.62** | **0.90** | **0.60** | **0.68** |
| TOFU Forget05 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.92 | 0.53 | 9.40e-17 | 0.97 | 0.46 | 0.93 | 0.63 | 0.90 | 0.57 | 0.64 |
| DPO | 0.82 | 0.57 | 9.20e-14 | 0.86 | 0.44 | 0.91 | 0.61 | 0.87 | 0.56 | 0.62 |
| NPO | 0.89 | 0.54 | 2.60e-16 | 0.95 | 0.46 | 0.94 | 0.63 | 0.90 | 0.57 | 0.64 |
| GA | 0.90 | 0.54 | 6.80e-16 | 0.96 | 0.46 | 0.94 | 0.63 | 0.90 | 0.57 | 0.64 |
| GD | 0.93 | 0.52 | 6.80e-16 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.56 | 0.64 |
| IHL | 0.94 | 0.52 | 7.00e-17 | 0.98 | 0.47 | 0.94 | 0.62 | 0.90 | 0.56 | 0.64 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.01 | 0.83 | 1.40e-15 | 0.01 | 0.10 | 0.00 | 0.17 | 0.00 | 0.23 | 0.00 |
| GD+FILA | 0.02 | 0.77 | 1.70e-08 | 0.03 | 0.14 | 0.01 | 0.17 | 0.00 | 0.21 | 0.00 |
| LoKU | 0.54 | 0.58 | 6.90e-13 | 0.90 | 0.45 | 0.92 | 0.62 | 0.89 | 0.60 | 0.64 |
| **OURS (GD+Sine)** | **0.32** | **0.55** | **5.1e-01** | **0.90** | **0.45** | **0.94** | **0.62** | **0.89** | **0.60** | **0.64** |
| TOFU Forget10 | | | | | | | | | | |
| *Full Fine-tuning Methods* | | | | | | | | | | |
| KL | 0.46 | 0.65 | 2.70e-05 | 0.47 | 0.35 | 0.93 | 0.55 | 0.89 | 0.56 | 0.49 |
| DPO | 0.44 | 0.55 | 5.30e-17 | 0.66 | 0.44 | 0.82 | 0.54 | 0.87 | 0.51 | 0.57 |
| NPO | 0.53 | 0.65 | 3.50e-06 | 0.54 | 0.35 | 0.94 | 0.50 | 0.90 | 0.51 | 0.47 |
| GA | 0.48 | 0.66 | 2.40e-03 | 0.49 | 0.33 | 0.93 | 0.51 | 0.91 | 0.50 | 0.39 |
| GD | 0.82 | 0.51 | 2.30e-16 | 0.92 | 0.47 | 0.92 | 0.60 | 0.88 | 0.55 | 0.62 |
| IHL | 0.73 | 0.57 | 3.80e-15 | 0.88 | 0.45 | 0.94 | 0.64 | 0.89 | 0.59 | 0.64 |
| *Parameter-Efficient Methods* | | | | | | | | | | |
| GA+FILA | 0.02 | 0.86 | 5.50e-18 | 0.02 | 0.09 | 0.00 | 0.19 | 0.00 | 0.18 | 0.00 |
| GD+FILA | 0.01 | 0.85 | 1.90e-21 | 0.01 | 0.09 | 0.00 | 0.18 | 0.00 | 0.18 | 0.00 |
| LoKU | 0.29 | 0.65 | 2.90e-01 | 0.91 | 0.45 | 0.89 | 0.62 | 0.88 | 0.57 | 0.63 |
| **OURS (GD+Sine)** | **0.32** | **0.53** | **8.7e-01** | **0.92** | **0.47** | **0.94** | **0.62** | **0.89** | **0.60** | **0.68** |

**Table 21:** Comprehensive **TOFU evaluation results for the Llama3.1-8B utilizing ranks {4, 8, 16, 32} LoRA for Parameter-Efficient Methods** across three forget splits (1%, 5%, 10% of authors) in accordance with the evaluation protocol outlined by [33]. "Original" denotes the pretrained model without any unlearning operations, whereas "Retain90" refers to a model retrained solely on 90% of the data (excluding the forget set) without implementing unlearning procedures. The metrics assessed included forget quality (FQ), model utility (MU), Rouge-L scores, and probability measures for both the forget and retain datasets. Superior unlearning performance is characterized by the highest FQ and MU values, low Rouge-L and probability scores on forget data, and high Rouge-L and probability scores on retain data, aligning with current literature [7].

| Method | Split | Forget Set | | FQ (↑) | Retain Set | | Real Authors | | Real World | | MU (↑) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | RL (↓) | TR (↓) | | RL (↑) | TR (↑) | RL (↑) | TR (↑) | RL (↑) | TR (↑) | |
| Original | – | 0.99 | 0.51 | 2.19e-20 | 0.98 | 0.47 | 0.94 | 0.62 | 0.89 | 0.55 | 0.63 |
| Retain90 | – | 0.40 | 0.67 | 9.50e-01 | 0.98 | 0.47 | 0.92 | 0.61 | 0.88 | 0.55 | 0.63 |
| ***Our Method: Performance Across LoRA Ranks*** | | | | | | | | | | | |
| **OURS (GD+Sine)** $r$=4 | forget01 | 0.40 | 0.50 | 8.50e-01 | 0.98 | 0.48 | 0.94 | 0.62 | 0.90 | 0.60 | 0.68 |
| | forget05 | 0.35 | 0.49 | 5.00e-01 | 0.97 | 0.47 | 0.94 | 0.62 | 0.89 | 0.60 | 0.64 |
| | forget10 | 0.31 | 0.50 | 8.30e-01 | 0.93 | 0.48 | 0.94 | 0.62 | 0.89 | 0.60 | 0.68 |
| **OURS (GD+Sine)** $r$=8 | forget01 | 0.40 | 0.50 | 8.90e-01 | 0.98 | 0.48 | 0.94 | 0.62 | 0.90 | 0.60 | 0.68 |
| | forget05 | 0.35 | 0.49 | 5.00e-01 | 0.97 | 0.47 | 0.94 | 0.62 | 0.89 | 0.60 | 0.64 |
| | forget10 | 0.31 | 0.50 | 8.00e-01 | 0.93 | 0.48 | 0.94 | 0.62 | 0.89 | 0.60 | 0.68 |
| **OURS (GD+Sine)** $r$=16 | forget01 | 0.40 | 0.50 | 8.90e-01 | 0.98 | 0.48 | 0.94 | 0.62 | 0.90 | 0.60 | 0.68 |
| | forget05 | 0.35 | 0.49 | 5.00e-01 | 0.97 | 0.47 | 0.94 | 0.62 | 0.89 | 0.60 | 0.64 |
| | forget10 | 0.31 | 0.50 | 8.00e-01 | 0.93 | 0.48 | 0.94 | 0.62 | 0.89 | 0.60 | 0.68 |
| **OURS (GD+Sine)** $r$=32 | forget01 | 0.40 | 0.50 | 8.50e-01 | 0.98 | 0.48 | 0.94 | 0.62 | 0.90 | 0.60 | 0.68 |
| | forget05 | 0.35 | 0.49 | 5.00e-01 | 0.97 | 0.47 | 0.94 | 0.62 | 0.89 | 0.60 | 0.64 |
| | forget10 | 0.31 | 0.50 | 8.00e-01 | 0.93 | 0.48 | 0.94 | 0.62 | 0.89 | 0.60 | 0.68 |

**Note:** RL = Rouge-L, TR = Truth Ratio. Our method consistently achieves stable performance across all LoRA ranks (4, 8, 16, 32) and forget splits (1%, 5%, 10%), demonstrating scalability and rank-agnostic effectiveness while preserving the model utility.