Interactive LoRA Steering: A User-Guided Framework for Efficient and Interpretable Machine Unlearning in Neural Networks

VAISHNAVEE SANAM[1], HERAMB PATIL[2], DR. MINAKSHI ATRE[1]
[1]Department of Artificial Intelligence and Data Science, PVGCOET&GKPIM, Pune, INDIA
[2]Department of Electronics and Telecommunication Engineering, PVGCOET&GKPIM, Pune, INDIA

Abstract: — The need to selectively remove information from trained machine learning models—a process termed Machine Unlearning—is critical for maintaining data privacy and model relevance. While exact unlearning via retraining is often prohibitively expensive, especially for large models, existing approximate methods can suffer from instability, catastrophic forgetting, or lack fine-grained control. This paper introduces Interactive LoRA Steering, a framework combining the parameter efficiency of Low-Rank Adaptation (LoRA) with stable unlearning objectives like Inverted Hinge Loss (IHL). Crucially, it incorporates a human-in-the-loop mechanism allowing users to guide the unlearning direction and intensity via conceptual "Steering Vectors" acting on LoRA adapters. We demonstrate through experiments on MNIST and CIFAR-10 that the core LoRA & IHL mechanism effectively and efficiently removes targeted information, significantly outperforming unstable baselines like Gradient Ascent while preserving utility on retained data and achieving favorable privacy metrics (MIA). We further show the conceptual feasibility of the interactive component, representing a step towards more controllable and interpretable "knowledge surgery" in AI systems. Our results show that LoRA+IHL achieves competitive utility on retained data (e.g., 92.6% Retain Acc on CIFAR-10), effectively reduces forget class accuracy (e.g., to 5.1% on CIFAR-10), and yields strong privacy metrics (e.g., 0.66 MIA Efficacy on CIFAR-10), performing comparably to retraining but completing the unlearning process significantly faster (e.g., ~10 seconds vs. ~155 seconds on CIFAR-10)

Keywords— machine unlearning, LoRA, Inverted Hinge Loss, knowledge surgery
Received: April 29, 2025. Revised: June 18, 2025. Accepted: July 6, 2025. Published: August 11, 2025.

1. Introduction

Modern deep learning algorithms are frequently taught on large datasets, which may contain sensitive user data. When such data must be removed—due to privacy restrictions such as the GDPR's "Right to be Forgotten" [3]—deleting information from storage is insufficient; its impact on the model must also be deleted. This approach, known as Machine Unlearning [1, 2], seeks to eliminate learnt correlations with specific data points.

While retraining from zero guarantees precise unlearning [1, 4], it is computationally expensive, particularly for large-scale models such as LLMs [7]. This has sparked interest in approximation approaches [2, 11], while many—such as those based on gradient ascent [8, 9]—can be unstable or incomplete [17, 20], while others remain unduly complicated or resource-intensive [10].

However, most techniques lack fine-grained, user-guided control. To address this, we propose Interactive LoRA Steering—a framework combining LoRA's efficiency [12], IHL's stability [17], and human-in-the-loop control via conceptual steering vectors. This enables semantically guided, transparent, and controllable unlearning. We present the framework, evaluate its performance, and explore trade-offs, advancing user-aligned, trustworthy AI.

2. Related Work

Machine unlearning, the process of removing the influence of specific data subsets from trained models, has become a crucial research area driven by privacy regulations like GDPR's "Right to be Forgotten" [3] and the need for model maintenance (e.g., removing harmful or outdated information) [2]. The primary challenge lies in achieving this removal efficiently without compromising the model's performance on the remaining data.

2.1. Exact vs. Approximate Unlearning

Unlearning methods are broadly categorized as exact or approximate [2]. Exact unlearning aims to produce a model statistically indistinguishable from one retrained from scratch on the retained data [1]. While providing strong guarantees, methods like SISA [4], which partition data and retrain sub-models, often incur significant computational or storage overhead, limiting practicality for large models. Consequently, research has increasingly focused on approximate unlearning [2, 11].

2.2 Approximate Unlearning Techniques and Challenges

These methods modify the parameters of the already trained model to approximate the state of a retrained model, prioritizing efficiency. Foundational approaches include:

- **Gradient-Based Methods:** Techniques like Gradient Ascent (GA or NegGrad) [8, 9] attempt to "reverse" learning by maximizing the loss on the data to be forgotten (forget set, Df). However, standard GA using losses like cross-entropy can suffer from optimization instability and catastrophic forgetting of retained knowledge [17, 20]. Methods trying to mitigate this by adding regularizing losses can still face stability issues.
- **Influence Function & Fisher Information Methods:** Other techniques leverage influence functions [10] or Fisher Information [8] to estimate the impact of specific data points on model parameters and attempt to counteract this influence. While principled, these can be computationally

intensive or complex to implement, especially for large models.

- **Privacy Risks:** A critical challenge for any approximate method is ensuring effective removal; imperfect unlearning may leave models vulnerable to Membership Inference Attacks (MIA) [5, 6], which aim to determine if specific data was part of the training set.

2.3 Efficiency Enhancements: Parameter-Efficient Fine-Tuning (PEFT) and LoRA

Recognizing the computational burden of updating large models, particularly in fine-tuning and adaptation contexts, Parameter-Efficient Fine-Tuning (PEFT) techniques have emerged. Low-Rank Adaptation (LoRA) [12] is a prominent PEFT method. LoRA freezes the base model weights and injects small, trainable low-rank matrices (adapters, where update $\Delta W = BA$) into specific layers. Training only these adapters drastically reduces the number of trainable parameters, improving computational efficiency and memory usage [12]. Furthermore, LoRA offers inherent regularization properties and better stability against catastrophic forgetting compared to full fine-tuning [13].

2.4 Recent State-of-the-Art in LoRA-Based Unlearning

The efficiency and modularity of LoRA make it an attractive candidate for approximate unlearning. Several recent studies have explored this:

- **NegLoRA [16]:** This work directly applies the negative gradient ascent principle (GA) to the LoRA adapters, training only the low-rank matrices to maximize loss on the forget set. It demonstrates efficiency gains but may inherit stability issues from standard GA.
- **PruneLoRA [14]:** This method combines model sparsification (pruning) with LoRA adaptation and subsequent unlearning, showing potential trade-offs between unlearning effectiveness, performance retention, and efficiency.
- **Residual Feature Alignment LoRA (RFA LoRA) [15]:** This approach uses LoRA adapters to align features of the forget set with the average features of the retain set, aiming for consistency at the intermediate representation level.
- **IHL & FILA for LLM Unlearning [17]:** Focused on large language models (LLMs), this work addresses GA instability by proposing the Inverted Hinge Loss (IHL) for more stable optimization during unlearning with LoRA. It also introduces Fisher-weighted Initialization for LoRA Adapters (FILA) to potentially accelerate the unlearning process. Our work leverages the stability insights from IHL [17].

2.5 Identified Gap: Need for Control and Interpretability

Despite rapid progress in efficient approximate unlearning using LoRA [14, 15, 16, 17], existing SOTA methods primarily focus on automating the process to achieve efficiency and performance preservation. There remains a significant gap in frameworks that provide fine-grained user control, interactivity, and semantic guidance over the unlearning process. Current methods generally lack mechanisms for users to specify *what aspects* of knowledge to forget (beyond specific data points or classes) or to interpret and steer the forgetting process dynamically based on feedback. Our work, Interactive LoRA Steering, aims to fill this gap by proposing an interactive, user-centric paradigm built upon efficient and stable LoRA-based unlearning primitives.

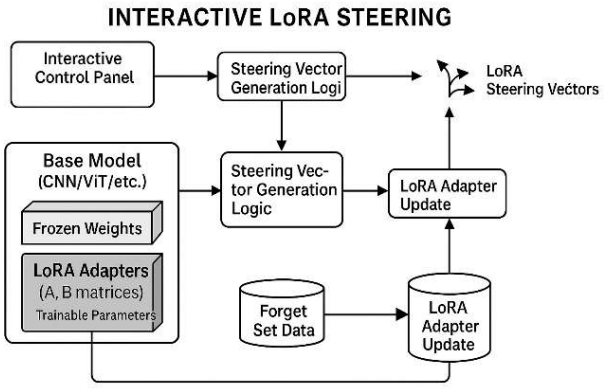
3. Proposed Framework

Current automated unlearning methods often lack the precise control needed for complex "knowledge surgery" tasks which go beyond simple data removal. To resolve this, we propose **Interactive LoRA Steering**, a framework designed to integrate user guidance directly into an efficient and stable unlearning process. Our goal is to transform unlearning from an opaque procedure into a controllable, interpretable technique by leveraging parameter efficiency and user interaction.

3.1 Overall Vision and Architecture

The core idea is to empower users to semantically guide the unlearning process in terms of both what specific concepts or data influences should be targeted (Direction) and how strongly they should be removed (Intensity). We achieve this by building upon the parameter efficiency of Low-Rank Adaptation [12] and the stability of unlearning methods like Inverted Hinge Loss (IHL) [17].

Fig 1. Architecture Diagram



The system comprises of a Base Model (e.g., CNN, ResNet, Transformer) whose original parameters (W) remain frozen during unlearning to preserve general knowledge and maintain efficiency. LoRA Adapters (low-rank matrices A, B where update $\Delta W = BA$) injected into selected layers. These adapters contain the only trainable parameters during the unlearning phase [12]. An Unlearning Mechanism (*Section B*) that updates the LoRA adapters based on the forget objective (e.g., IHL) applied to the target forget data (Df_target). An Interactive Control component (*Section D*) where user inputs (Target, Intensity, Semantic Direction) are taken guiding the Unlearning Mechanism.

3.2 Unlearning Mechanism: Targeted Adapter Updates via IHL

The mechanism for inducing forgetting operates solely by modifying the parameters of the LoRA adapters (A and B). We use the *Inverted Hinge Loss* (IHL) [17] calculated over the user-specified forget data (D_{f_target}). For a target output x_t (potentially conditioned on context $x_{<t}$), IHL is defined as:

$$L_{\text{IHL}}(x) = \max\left(0, m + p_\theta(x_t|x_{<t}) - \max_{v \neq x_t} p_\theta\,(v|x_{<t})\right)$$

Here:

- $p_\theta(x_t \mid x_{<t})$ is the model's predicted probability for token y given context $x_{<t}$, incorporating the effect of active LoRA adapters.
- x_t is the target token to forget, and $v \neq x_t$ denotes alternative vocabulary tokens.
- m is a positive margin hyperparameter controlling the separation between x_t and the most likely alternative token.

Negative cross-entropy used in standard Gradient Ascent (GA) gives decent results but IHL offers crucial advantages: its bounded nature prevents numerical instability and gradient explosion observed in GA [17], and it promotes targeted forgetting by directly penalizing the target output probability relative to the most likely alternative, potentially causing less collateral damage to the model's overall distribution. During unlearning, gradients of LIHL are computed only with respect to the trainable LoRA parameters (A, B) and used to update them via an optimizer like AdamW [24]. While standard LoRA initialization is used here, exploring Fisher-weighted initialization [17] is a potential future enhancement.

3.3 Interactive Steering of the Unlearning Process

The central novelty of our framework lies in enabling user interaction to guide the LoRA adapter updates. Via a conceptual control interface, users define:

- ***Target (D_{f_target})***: The specific data points or criteria defining the information to be forgotten.
- ***Intensity (I)***: The desired strength or depth of the unlearning effect.
- ***Semantic Direction (SD)***: High-level guidance on what features, concepts, or data characteristics within the Target set should be preferentially unlearned.

LoRA Steering Vector Logic interprets this user intent. This logic functions as a translator, transforming high-level directives into concrete configurations for the IHL-based unlearning process. For example, 'Intensity' may refer to LoRA rank r, IHL margin m, or the number of unlearning epochs; 'Semantic Direction' could be dynamically picking subsets of D_{f_target} or applying alternate weights during the IHL update based on data attributes. These "vectors" represent this control logic, not direct model parameters.

This enables an iterative workflow, outlined in Algorithm 1. The user sets initial parameters, the system performs an unlearning step on the LoRA adapters, feedback (e.g., performance metrics) is provided, and the user can then choose to stop, continue, or adjust the steering parameters (Intensity, Direction) for the next iteration.

ALGORITHM 1: INTERACTIVE LORA STEERING CYCLE

```
Input:    M (Frozen base model with active LoRA adapters A, B)
          D (Full dataset)
Output:  Updated LoRA adapters A, B
1:  settings ← InitUserSettings() // Target, Intensity, Direction
2:  loop
3:   // Configure: Translate user settings -> unlearning parameters
4:    Df_target, config ← Translate(settings, D) // Get forget data, epochs, rank, etc.
5:
6:  // Unlearn: Update LoRA adapters using IHL
7:     for epoch = 1 to config.epochs
8:           Update A, B using IHL_Loss(M, Df_target) // Gradients only for A, B
9:     end for
10:
11:    // Feedback: Evaluate performance & present to user
12:    metrics ← Evaluate(M, Df_target, ...) // Forget/Retain Acc
13:    PresentFeedback(metrics)
14:
15:        // Decide: Get user action & potentially new settings
16:         action,      new_settings      ← Get_User_Decision()   //   STOP,   CONTINUE, ADJUST
17:
18:    if action == STOP then
19:        break // Exit loop
20:    else if action == ADJUST then
21:        settings ← new_settings
22:    end if
23:     // Implicitly CONTINUE if action is neither STOP nor ADJUST
24: end loop
25: return A, B
```

This human-in-the-loop approach facilitates targeted, adaptable knowledge modification, offering a level of control and potential interpretability beyond that of fully automated methods.

4. Experimental Setup

This section talks about the methodology used to evaluate the proposed Interactive LoRA Steering framework, focusing on the core LoRA+IHL unlearning mechanism. Our objectives were: (1) assess the effectiveness, efficiency, and stability of LoRA+IHL against relevant baselines; (2) evaluate its impact on model utility and privacy (via MIA); and (3) demonstrate the conceptual feasibility of the interactive steering aspect.

4.1 Datasets and Tasks

We used standard image classification benchmarks:

1) ***MNIST*** [22]: Handwritten digits (0-9). Task: Forget class '7'.
2) ***CIFAR-10*** [23]: 10-class color images. Task: Forget class '3' ('cat').

For each dataset, the training set was split into a Forget Set (D_f), comprising all training samples of the target class (e.g., all 'cat' images from CIFAR-10 train), and a Retain Set (D_r), containing all remaining training samples. Standard dataset-specific normalization was applied. For Membership Inference Attack (MIA) evaluation, balanced datasets were created using samples from D_r (members, label 1) and the test set (non-members, label 0), following standard practice [5, 6].

4.2 Models and LoRA Configuration

1) ***MNIST***: A standard Convolutional Neural Network (CNN) with two convolutional layers followed by max-pooling and two linear layers (details in Appendix if space permits).

2) ***CIFAR-10***: A ResNet-18 architecture [21], initialized with ImageNet pre-trained weights and then fully fine-tuned on the complete CIFAR-10 training set to establish a realistic starting point (the 'Baseline' model).

For methods involving LoRA (LoRA FT, LoRA+GA, LoRA+IHL), adapters were injected into the final two linear layers of the classifier head. Unless otherwise specified for ablation studies, we used a default LoRA rank r=8, alpha=16, and dropout=0.1. Critically, the base model parameters remained frozen during all LoRA-based unlearning or fine-tuning steps; only adapter weights were updated. We utilized the Hugging Face PEFT library [25] for LoRA implementation.

4.3 Methods Compared

We evaluated our proposed LoRA+IHL approach against the following methods:

1) ***Baseline***: The original model trained on the full dataset ($D_r \cup D_f$).
2) ***Retraining***: Model trained from scratch only on the Retain Set (D_r); represents the theoretical 'gold standard' for forgetting.
3) ***GA (Full)***: All parameters in the baseline model undergo Standard Gradient Ascent with negative cross-entropy on D_f.
4) ***Parameter Zeroing***: The naïve strategy involves zeroing the weights and biases associated with the forget class in the final linear layer of the Baseline model.
5) ***LoRA FT (Retain)***: Baseline model with LoRA adapters optimised for D_r and standard cross-entropy loss.
6) ***LoRA+GA***: Gradient Ascent (negative cross-entropy) upgrades D_f's baseline model and LoRA adapters.

4.4 Evaluation Metrics

Performance was assessed using metrics covering utility, unlearning effectiveness, efficiency, and privacy:

1) ***Utility***: Accuracy on the full Test Set (Test Acc) and on test samples from retained classes (Retain Acc).
2) ***Forgetting***: Accuracy on test samples from the forget class (Forget Acc - lower is better).
3) ***Efficiency***: Wall-clock time (s) for the unlearning/retraining procedure.
4) ***Privacy (MIA)***: A Logistic Regression MIA predictor trained on model prediction confidences (retain vs. test set). Evaluated by:
5) ***MIA AUC***: Area Under the ROC Curve (closer to 0.5 indicates better privacy/less distinguishability).
6) ***MIA Efficacy***: Accuracy of the MIA predictor in classifying original Df samples as non-members (higher indicates Df samples look less like members after unlearning).

Steering (Interactive Demo):

7) ***Steering Accuracy***: Defined as ***2) Forget Accuracy***
8) ***Semantic Selectivity***: Qualitative assessment of accuracy drop on target vs. retained classes.

4.5 Implementation Details

Experiments were conducted using PyTorch [26] on NVIDIA Tesla T4 GPUs. We primarily used the AdamW optimizer [24]. Key hyperparameters, such as learning rates, number of epochs, the IHL margin m (typically 1.0 or 2.0), and gradient clipping for GA methods (clip=1.0), are noted with the results where applicable or detailed in the Appendix.

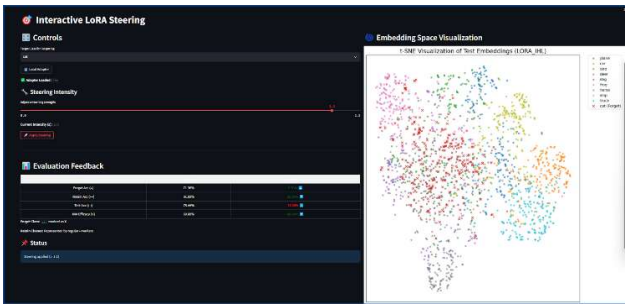
We created an interactive LoRA steering interface (Fig. 2) to show usability and practical feasibility, allowing users to conceptually and visually direct the unlearning process. Three primary inputs are supported by the system:

- **Target Class**: Chooses which data or class should be forgotten.
- **The steering intensity**, which is related to LoRA parameters like rank r, margin m, and training epochs, regulates how aggressively the model unlearns.
- Optional **semantic direction** gives precise conceptual direction on which aspects of the target should be removed.

In order to facilitate qualitative study of the effectiveness of unlearning, a live embedding visualization panel (right) uses t-SNE projections to demonstrate how the representation of the forgotten class disperses over time and assists in tracking semantic shifts after unlearning.

Users obtain immediate feedback via an evaluation table displaying updated metrics, including Test Accuracy, Forget Accuracy, and Retain Accuracy. This repeated unlearning process allows the user to modify steering parameters and promptly observe the effects. This loop exemplifies the human-in-the-loop design that supports our proposed technique.

Fig. 2. Schematic showing example user inputs



5. Results and Analysis

We evaluated the proposed Interactive LoRA Steering framework by analyzing the performance of its core LoRA+IHL unlearning mechanism against baselines on MNIST and CIFAR-10. Key results are summarized in Table 1.

TABLE I. COMPARATIVE EVALUATION OF UNLEARNING METHODS ON MNIST AND CIFAR-10

Method	Dataset	Test Acc ↑	Retain Acc ↑	Forget Acc ↓	MIA Efficacy ↑	Time (Unlearn) ↓
Baseline	*MNIST*	**0.9933**	0.9972	99.63 %	0.0826	N/A
	CIFAR-10	**0.7544**	0.9192	71.36 %	0.606	N/A
Retrained	*MNIST*	0.8904	0.9955	**0.00 %**	**0.7248**	53.73s

	CIFAR-10	0.7201	0.9228	**0.00%**	**0.6704**	155.55s
GA (Full)	*MNIST*	0.3522	0.3968	**0.00%**	0.9996	76.71s
	CIFAR-10	0.5742	0.6884	17.14%	0.543	94.99s
Zeroed	*MNIST*	0.9272	0.9974	34.49%	0.9158	**~0s**
	CIFAR-10	0.7191	0.9265	**0.70%**	0.6274	**~0s**
LoRA FT (Retain)	*MNIST*	0.9	**0.9979**	9.11%	0.8594	26.29s
	CIFAR-10	**0.7262**	**0.9385**	**0.00%**	0.606	77.41s
LoRA+IHL (Ours)	*MNIST*	**0.8924**	0.9974	**0.02%**	**0.8468**	**11.25s**
	CIFAR-10	**0.7218**	0.9266	**5.12%**	**0.6636**	**10.31s**

5.1 Baseline Performance Analysis

The baseline methods established important benchmarks (Table 1). Retraining effectively reduced Forget Accuracy (near 0%) and achieved good MIA Efficacy (MNIST: 0.72, CIFAR-10: 0.67), confirming its status as a theoretical optimum for removal. However, it was the most time-consuming (MNIST: 53.7s, CIFAR-10: 155.6s) and incurred a noticeable drop in overall Test Accuracy compared to the original Baseline model (MNIST: 89.0% vs 99.3%, CIFAR-10: 72.0% vs 75.4%), highlighting potential utility costs. GA (Full) proved very unstable, causing catastrophic forgetting on MNIST (Test Acc: 35%) and failing to unlearn effectively in addition to still damaging utility on CIFAR-10 (Forget Acc: 17%, Test Acc: 57%). This underscores the unreliability of naive gradient ascent. Faster methods like Parameter Zeroing and LoRA FT (Retain) reduced Forget Accuracy significantly with minimal time cost, but their relatively poor MIA Efficacy scores (e.g., CIFAR-10: 0.63, 0.61 respectively) suggest the unlearning might be superficial compared to Retraining. These baseline results motivate the need for a method balancing effectiveness, efficiency, stability, and principled unlearning.

5.2 Core Mechanism Validation: LoRA+IHL vs. LoRA+GA

Our proposed LoRA+IHL method demonstrated a strong balance of properties (Table 1). Compared to LoRA+GA, LoRA+IHL exhibited significantly better stability. On MNIST, LoRA+GA collapsed (Test Acc: ~11%), mirroring the full GA instability, while LoRA+IHL maintained high Test/Retain accuracy (~89.2%/~99.7%) comparable to Retraining, alongside excellent forgetting (Forget Acc: 0.02%). On CIFAR-10, LoRA+IHL also preserved Test Accuracy slightly better than LoRA+GA (72.2% vs 71.7%) while achieving effective forgetting (Forget Acc ~5.1%).

In terms of efficiency, both LoRA+IHL and LoRA+GA were very fast, updating only the small adapter sets using the forget data (MNIST: ~11s, CIFAR-10: ~9-10s) – a substantial improvement over Retraining or even LoRA FT on the larger retain set.

Crucially, LoRA+IHL demonstrated superior performance regarding privacy metrics. Its MIA Efficacy (MNIST: 0.85, CIFAR-10: 0.66) and MIA AUC scores were consistently closer to the Retraining benchmark than those of LoRA+GA (CIFAR-10 MIA Efficacy: 0.62). This suggests that IHL promotes a more fundamental removal of the forget set's influence, making it harder for an MIA predictor to distinguish forgotten samples, whereas GA's effect might be less targeted. Overall, LoRA+IHL provides an effective, efficient, and stable unlearning mechanism with favorable privacy characteristics.

5.3 Interactive Feasibility Demonstration

We simulated the interactive workflow using LoRA+IHL on CIFAR-10 (forgetting 'cat'). The simulation confirmed feasibility: the user conceptually selected the target and intensity (mapped to rank 8, 10 epochs), triggering the efficient (~10s) LoRA+IHL unlearning step. Results showed high Steering Accuracy (1 - Forget Acc $\approx$ 94.9%) and good Semantic Selectivity, with a large accuracy drop (~74% absolute) on the target 'cat' class but only minimal impact on retained classes (Test Acc. dropped slightly from 75.4% to 72.2%). Primary tests changing LoRA rank also confirmed the ability to trade off forgetting depth vs. utility preservation, supporting the potential for user control over intensity.

5.4 Visualization Analysis

Qualitative investigation employing t-SNE visualizations of penultimate layer embeddings (Figure 2, CIFAR-10) backs up these findings. In contrast to the Baseline model's obvious class clusters, the Retrained model demonstrates that the target 'cat' cluster has been dissolved. The LoRA+IHL visualization is very similar to the Retrained one: the 'cat' cluster is greatly dispersed, while the clusters for retained classes are mostly intact. This visibly validates the targeted unlearning and selectivity found numerically, in contrast with the anticipated larger distortions expected from unstable approaches such as GA.

6. Discussion and Future Work

The results presented validate the core LoRA+IHL mechanism of our proposed Interactive LoRA Steering framework. Our findings demonstrate that this combination outperforms unstable baselines like Gradient Ascent [8, 9] and offers a better balance of effectiveness, efficiency, and privacy preservation compared to naive or simple PEFT approaches on MNIST and CIFAR-10. This section discusses these results in the context of state-of-the-art methods, outlines the novelty and limitations of our interactive approach, and identifies future research directions.

6.1 Core Mechanism Performance and Comparison to SOTA Concepts

Our LoRA+IHL mechanism [12, 17] proved effective and efficient (Table 1). It significantly outperformed full Gradient Ascent (GA) in terms of stability and utility preservation, avoiding the catastrophic forgetting observed in GA (Full). Compared to simple LoRA fine-tuning on the retain set (LoRA FT (Retain)), our method achieved comparable or better forgetting with potentially improved privacy metrics (MIA Efficacy closer to Retraining), suggesting a more principled removal than simply optimizing for retained data.

While direct numerical comparison with all recent LoRA-based SOTA methods like NegLoRA [16], PruneLoRA [14], and RFA LoRA [15] was beyond the scope of this initial study due to implementation complexities and focus, our LoRA+IHL component conceptually aligns with the goals of

stability and efficiency highlighted in works like Cha et al. [17]. The use of IHL specifically addresses the instability noted in GA-based methods [17], including potentially NegLoRA [16] if it relies on standard negative losses.

6.2 Novelty: User Control, Semantic Guidance, and Interpretability

The primary novelty of Interactive LoRA Steering lies not just in the core unlearning algorithm (which leverages LoRA [12] and IHL [17]) but in the proposed **interactive, user-centric paradigm**:

- **User Control & Interactivity:** Unlike fully automated SOTA methods [14, 15, 16, 17], our framework introduces a human-in-the-loop mechanism. Users can initiate, monitor (via feedback like metrics and visualizations, Fig. 2), guide (Intensity, Target), and iteratively refine the unlearning process. This transforms unlearning from a one-shot black-box operation into a controllable tool.
- **Semantic Guidance & "Knowledge Surgery":** Current SOTA typically targets data points or classes. Our framework conceptually allows users to control the *semantic direction* of forgetting via "Steering Vectors," moving beyond data removal towards more nuanced "knowledge surgery" (e.g., forgetting specific attributes or concepts within the target data).
- **Enhanced Interpretability:** The interactive nature, combined with potential visualizations (like the t-SNE embeddings in Fig. 2), provides greater transparency. Users can observe the effects of their steering actions, gaining insights into how the model's knowledge representation is modified during unlearning.

6.3 Limitations of the Current Work

It is crucial to acknowledge the limitations:

- **Preliminary Empirical Validation:** The experiments focused on validating the core LoRA+IHL mechanism against basic baselines on image classification tasks (MNIST, CIFAR-10). Comprehensive benchmarking against the full suite of recent SOTA methods [14, 15, 16, 17] across diverse tasks (especially NLP/LLMs) and unlearning scenarios is necessary future work.
- **Conceptual Steering Mechanism:** The "Steering Vector Logic" translating high-level user intent (Intensity, Semantic Direction) into low-level unlearning configurations (e.g., IHL margin, LoRA rank, subset selection) is currently conceptual. Implementing robust and generalizable logic, potentially through learned components, is a significant challenge.
- **User Interface Simulation:** The interactive component was simulated programmatically (Algorithm 1) and conceptually visualized (Fig. 2). A full implementation requires careful UI/UX design, and user studies are needed to assess its practical usability and effectiveness.
- **Scalability to LLMs:** While motivated by challenges in large models [7] and using efficient techniques [12], the framework's scalability and effectiveness for models with billions of parameters remain unproven and require dedicated investigation.
- **Approximate Nature:** Like most efficient unlearning methods [2, 11, 14-17], our approach is approximate and does not offer the formal guarantees of exact retraining [1, 4].

6.4 Future Work

These limitations point to key future directions:

1. **Extend and Benchmark on LLMs:** Adapt and rigorously evaluate Interactive LoRA Steering on NLP tasks and LLMs, comparing against SOTA unlearning techniques [14-17].
2. **Develop Steering Vector Logic:** Research methods (rule-based, learning-based) to translate user intent into effective LoRA adapter updates and IHL configurations. Explore mapping semantic goals (e.g., "forget toxicity," "remove bias related to X") to parameter-level changes.
3. **Implement and Evaluate Interactive Interface:** Build a functional prototype and conduct user studies to assess the usability and utility of the interactive controls.
4. **Explore Algorithmic Refinements:** Investigate adaptive LoRA ranks, sophisticated initialization strategies like FILA [17], and integration with other PEFT methods or knowledge editing techniques.
5. **Theoretical Analysis:** Explore theoretical bounds or properties related to the interactive unlearning process, potentially connecting user actions to changes in model behaviour or privacy characteristics.

In summary, Interactive LoRA Steering proposes a shift towards user-controlled, interpretable unlearning. While leveraging efficient primitives like LoRA [12] and stable objectives like IHL [17], its main contribution is the interactive paradigm. Significant future work is needed to fully realize its potential, particularly in steering logic design and large-model scalability.

7. Conclusion

This paper introduced Interactive LoRA Steering, a framework designed to enhance the efficiency, stability, and user control of machine unlearning. By combining the parameter efficiency of LoRA [12] with the stability of objectives like Inverted Hinge Loss [17], and coupling this with a conceptual human-in-the-loop steering mechanism, we aim to move unlearning from automated bulk removal towards more precise, interpretable, and user-guided knowledge modification. Our findings demonstrate that the core LoRA+IHL unlearning technique is effective and significantly more stable and efficient than standard baselines like full gradient ascent [8] or computationally expensive retraining [1, 4]. The preliminary feasibility demonstration of the interactive component suggests a promising direction for addressing the limitations in controllability and interpretability of current automated SOTA methods [14, 15, 16, 17]. While further development is required, particularly in refining the steering logic and scaling to LLMs [7], Interactive LoRA Steering represents a potential step towards more

trustworthy, governable, and adaptive AI systems where users can actively shape model knowledge.

References

[1] Y. Cao and J. Yang, "Towards Making Systems Forget with Machine Unlearning," Proc. IEEE Symp. Security and Privacy (SP), 2015, pp. 463–480.

[2] H. Xu, T. Zhu, L. Zhang, W. Zhou, and P. S. Yu, "Machine Unlearning: A Survey," ACM Comput. Surv., vol. 56, no. 1, pp. 9:1–9:36, 2024. (Or cite another recent survey like Nguyen et al. 2022)

[3] P. Voigt and A. Von dem Bussche, The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing, 2017. (Or cite Mantelero 2013 / Rosen 2011 for GDPR/"Right to be Forgotten" context).

[4] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine Unlearning," Proc. IEEE Symp. Security and Privacy (SP), 2021, pp. 141–159. (Introduced SISA).

[5] M. Chen, Z. Zhang, T. Wang, M. Backes, M. Humbert, and Y. Zhang, "When Machine Unlearning Jeopardizes Privacy," Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS), 2021, pp. 896–911.

[6] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks against Machine Learning Models," Proc. IEEE Symp. Security and Privacy (SP), 2017, pp. 3-18. (Foundation for MIA).

[7] N. Carlini, F. Tramer, E. Wallace, et al., "Extracting training data from large language models," Proc. USENIX Security Symposium, 2021.

[8] A. Golatkar, A. Achille, and S. Soatto, "Eternal sunshine of the spotless net: Selective forgetting in deep networks," Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9304–9312.

[9] J. Jang, D. Yoon, S. Yang, S. Cha, M. Lee, L. Logeswaran, and M. Seo, "Knowledge unlearning for mitigating privacy risks in language models," Proc. Annual Meeting of the Association for Computational Linguistics (ACL), 2023..

[10] C. Guo, T. Goldstein, A. Hannun, and L. Van Der Maaten, "Certified data removal from machine learning models," Proc. Int. Conf. Machine Learning (ICML), 2020.

[11] S. Neel, A. Roth, and S. Sharifi-Malvajerdi, "Descent-to-delete: Gradient-based methods for machine unlearning," Proc. Algorithmic Learning Theory (ALT), 2021.

[12] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," Int. Conf. Learning Representations (ICLR), 2022.

[13] D. Biderman, J. Portes, J. J. G. Ortiz, et al., "LoRA Learns Less and Forgets Less," Transactions on Machine Learning Research (TMLR), 2024.

[14] A. Mittal, "LoRA Unlearns More and Retains More (Student Abstract)," ArXiv preprint arXiv:2411.11907, 2024. (Introduced PruneLoRA, uses NegLoRA.

[15] L. Qin, T. Zhu, L. Wang, and W. Zhou, "Machine Unlearning on Pre-trained Models by Residual Feature Alignment Using LoRA," ArXiv preprint arXiv:2411.08443, 2024. (Proposed RFA LoRA).

[16] J. Yu, W. Liu, D. Li, and P. Wang, "NegLoRA: Parameter-Efficient Deep Unlearning based on Low-Rank Adaptation," arXiv preprint arXiv:2404.04866, 2024. [Online]. Available: https://arxiv.org/abs/2404.04866

[17] S. Cha, S. Cho, D. Hwang, and M. Lee, "Towards Robust and Efficient Parameter-Efficient Knowledge Unlearning for Large Language Models," arXiv preprint arXiv:2310.00787, 2023. [Online]. Available: https://arxiv.org/abs/2310.00787

[18] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," Journal of Machine Learning Research (JMLR), vol. 9, pp. 2579-2605, 2008.

[19] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," ArXiv preprint arXiv:1802.03426, 2018. (If using UMAP).

[20] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, et al., "Overcoming catastrophic forgetting in neural networks," Proc. National Academy of Sciences (PNAS), vol. 114, no. 13, pp. 3521–3526, 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[23] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., University of Toronto, 2009.

[24] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," arXiv preprint arXiv:1711.05101, 2019.

[25] Hugging Face team, Parameter-Efficient Fine-Tuning (PEFT) library. Available: https://github.com/huggingface/peft. [Accessed: Mar, 2025].

[26] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems 32 (NeurIPS), 2019, pp. 8026–8037.