Low Rank Adaptations for Effective Machine Unlearning

Anonymous submission

Abstract

Growing privacy regulations have made machine unlearning an essential process for removing the influence of specific data points from trained models. While retraining on the remaining dataset is a straightforward solution, it incurs high computational costs and requires access to the retained dataset, which may not always be practical. Existing unlearning methods, such as gradient ascent, often suffer from unstable optimization and catastrophic forgetting. Recent studies have demonstrated that by training fewer parameters, Low-Rank Adaptation (LoRA) constrains updates to prevent significant divergence from the base model, effectively mitigating catastrophic forgetting. Building on this insight, we propose a novel framework, NegLoRA that leverages LoRA to enhance the efficiency and effectiveness of machine unlearning. Experimental results across various metrics indicate that NegLoRA outperforms baseline methods in unlearning accuracy, generalization, and robustness to inference attacks while being computationally efficient. Our code is available at https://github.com/AAAI-ColorAI/NegLoRA

Introduction

Deep Neural Networks have demonstrated remarkable accuracies across numerous tasks by harnessing vast datasets. However, this success brings the challenge of data privacy, as these models may inadvertently memorize training examples, making them vulnerable to inference attacks that could reveal sensitive information potentially harming user privacy. Privacy regulations like the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) grant individuals the "Right to be forgotten" allowing them to request the removal of their data from models and services. To address these concerns, Machine Unlearning methods systematically remove data points from a model, ensuring that once the data has been erased the model behaves as though it had never encountered that information.

The naive but optimal method method involves training the model from scratch using the remaining training set, after removing the data points to be forgotten. This procedure, termed "exact unlearning", guarantees that the weights of the resulting model aren't influenced by the instances to forget, but proves to be impractical for large-scale models and datasets due to substantial computational costs. Moreover, loading the original dataset may be unfeasible due to data retention policies or limitations in storage capacity for large datasets. These limitations entail the need for efficient unlearning methods given access only to the pre-trained model and the data points requested for deletion.

Several methods have been proposed to address this challenge. (Graves, Nagisetty, and Ganesh 2020) introduced a technique that stores the index of training examples and per-batch parameter updates, and unlearns by applying the stored gradients in the opposite direction, effectively reversing the influence of the data point on the model parameters. However, this requires extensive storage and may lead to a model that poorly approximates the one that would have existed if those parameter updates had not been applied.

To reduce storage needs and computational overhead, (Golatkar, Achille, and Soatto 2020) introduce NegGrad, which applies negative gradient updates for the samples to be forgotten, equivalently moving in the direction of increasing loss for those samples. This approach aims to hinder the model's ability to classify the "forget" samples correctly. However, it often leads to unstable training objectives and can lead to catastrophic forgetting, impairing the model's generalizability on the primary task by inadvertently erasing useful features learned from other data. Moreover, recent studies (Kurmanji et al. 2023) have shown that diverging algorithms such as gradien ascent, can cause uncharacteristially high errors on deleted examples making the model susceptible to Membership Inference Attacks.

Recent studies such as (Hu et al. 2021) show that model adaptation intrinsically involves low-rank rank changes in its weights. Building on this conclusion, Low Rank Adaptation (LoRA) freezes the pretrained model's weights and injects trainable rank decomposition matrices to reduce the number of parameters updated during fine-tuning, which significantly decreases computational costs. Furthermore, by training fewer parameters LoRA constraints the finetuned model from diverging significantly from the base model (Biderman et al. 2024), effectively acting as a regularizer that mitigates catastrophic forgetting more than other regularization techniques, while achieving similar or even better performance than full fine-tuning in many cases.

Building on LoRA's parameter efficiency and regularization benefits, we propose NegLoRA, an unlearning method which leverages Low-Rank Adaptation to apply targeted negative gradient updates. By efficiently guiding the model's parameters away from the influence of specific data points,

NegLoRA ensures effective forgetting while preserving high accuracy on both retained and test datasets. This approach also significantly reduces computational and memory demands through parameter-efficient low-rank updates, making it a scalable and effective solution for unlearning.

To further enhance efficiency, (Jia et al. 2024) leverages the Lottery Ticket Hypothesis (Frankle and Carbin 2019) to introduce sparsity by pruning specific weights or neurons in the trained model before unlearning, reducing overfitting and computational costs. However, applying gradient ascent to sparse models leads to a steep decline in the model's accuracy as sparsity increases. To address this, we propose Sparse NegLoRA using LoRA to unlearn in sprase models which mitigates the adverse effects of gradient ascent, effectively balancing unlearning efficiency, computational cost, and maintaining overall model performance.

Background and Notation

LoRA

Low-Rank Adaptation (LoRA) (Hu et al. 2021) is a technique for efficiently fine-tuning large models by introducing low-rank parameter updates, reducing computational and memory costs. Instead of updating the entire weight matrix $W \in \mathbb{R}^{d\times k}$, LoRA factorizes the update as $W' = W + \Delta W$, where ΔW is approximated by the product of two low-rank matrices $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ with rank $r \ll \min(d, k)$. This can be expressed as:

$$W' = W + BA,$$

where only the matrices A and B are trained, while W remains fixed. During fine-tuning, LoRA only modifies A and B, allowing the model to adapt to new data with minimal parameter adjustments, which avoids overfitting and reduces catastrophic forgetting of previously learned information. By introducing low-rank updates, LoRA achieves an efficient, scalable adaptation, making it particularly suitable for applications requiring continual learning or unlearning, as it focuses parameter updates on specific subspaces of the model.

Unlearning

Let $\phi(\cdot) \in \mathbb{R}^K$ be a model, with parameters w (weights) trained using dataset D. The k-th component of the vector $\phi(x; w)$ in response to an image x approximates the log-posterior, i.e., $\phi(x; w)_k \approx \log P(y = k|x)$,, up to a normalizing constant.

Given a model $\phi(x; w)$, where x is an input and w represents the model parameters, we aim to selectively unlearn a subset of data $D_f \subset D$ while retaining knowledge from $D_r = D \setminus D_f$. After training on dataset D, the model parameters are denoted as w_D, resulting in a model $\phi(x; w_D)$. Our goal is to design a scrubbing function S that modifies w_D to produce updated parameters $S(w_D)$, such that $\phi(x; S(w_D)) \approx \phi(x; w_{D_r})$, effectively removing the influence of D_f and mimicking a model trained only on D_r.

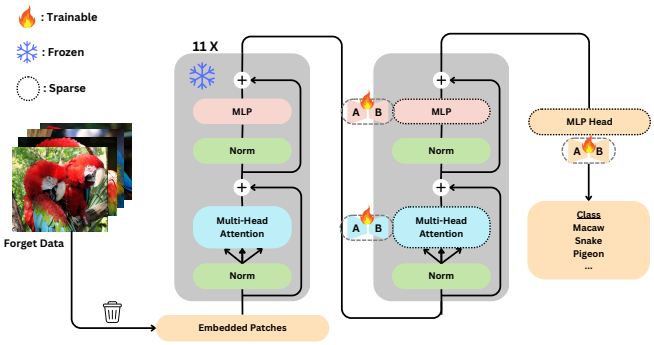

Figure 1: Sparse NegLoRA on ViT

Methodology

We focus on the problem of class unlearning in multi-class classification tasks, where the aim is to remove information about a specific class, called the forget class, from a trained model.

NegGrad

GA: (Graves, Nagisetty, and Ganesh 2020) (Thudi et al. 2022) (Golatkar, Achille, and Soatto 2020) We define a loss function $\mathcal{L}(\phi(x; w), y)$ on D_f that quantifies the model's performance on the data to be forgotten. Our objective is to maximize this loss to reduce the model's reliance on D_f. Here, the loss function $\mathcal{L}$ for unlearning can be defined as:

$$\mathcal{L}(\phi(x; w), y) = -\frac{1}{|D_f|} \sum_{(x,y)\in D_f} \log P(y|x; w),$$

where $P(y|x; w)$ is the model's predicted probability for the correct label y given input x. We perform iterative updates to w_D using gradient ascent:

$$w \leftarrow w + \eta \nabla_w \mathcal{L}(\phi(x; w), y),$$

where η is the learning rate. $\phi(x; S(w_D))$, approximates $\phi(x; w_{D_r})$, aligning with privacy requirements for data removal.

NegLoRA

While the vanilla NegGrad approach is successful in unlearning D_f, it frequently leads to reduced performance on D_r, decreasing the model's overall utility. (Zhang et al. 2024) determine that gradient ascent approaches can lead to *catastrophic collapse* of the model due to the divergent nature of the algorithm as well as the unbounded nature of its loss function. To address this, we propose NegLoRA, which exploits the fundamental regularization properties of Low Rank Adaptations by following a *first adapt, then ascend* paradigm. By training fewer parameters LoRA constrains the fine-tuned model from diverging significantly from the base model. By prioritizing "forgetting" in our loss function and using low-rank updates, NegLoRA achieves higher unlearning accuracy while maintaining model performance

Model	# Params	Unlearning Accuracy (UA)		MIA-Efficacy		Remaining Accuracy (RA)		Testing Accuracy (TA)		RTE
		1 Epoch	2 Epochs	1 Epoch	2 Epochs	1 Epoch	2 Epochs	1 Epoch	2 Epochs	(/epoch)
ImageNet100										
Retrain	21.7M	100.00	100.00	100.00	100.00	62.07	62.07	55.48	55.48	1.0000
GA(full)	21.7M	97.34	0.00	84.45	85.23	54.28	39.07	47.99	34.75	0.0081
GA (headatt)	1.86M	97.19	100.00	84.06	85.08	54.50	39.56	48.42	35.22	0.0081
GA (LoRA)	20K	100.00	100.00	96.80	97.27	**61.98**	60.62	**55.37**	54.38	0.0056
CIFAR100										
Retrain	21.7M	100.00	100.00	100.00	100.00	97.98	97.98	60.37	60.37	1.0000
GA (full)	21.7M	100.00	100.00	100.00	100.00	83.13	58.20	44.78	33.31	0.025
GA (headatt)	1.86M	100.00	100.00	100.00	100.00	95.94	94.98	55.77	49.15	0.025
GA LoRA	20K	98.86	100.00	100.00	100.00	**96.87**	95.80	**57.76**	50.43	0.017
CIFAR10										
Retrain	21.7M	100.00	100.00	100.00	100.00	96.90	96.90	85.92	85.92	1.0000
GA (full)	21.7M	97.19	100.00	100.00	100.00	54.50	39.56	48.42	35.22	0.025
GA (headatt)	1.86M	97.50	99.88	100.00	100.00	96.22	92.80	84.80	79.10	0.025
GA LoRA	20K	93.57	100	100.00	100.00	**96.53**	96.50	85.01	**85.38**	0.017

Figure 2: Results of ViT When Tested Using Various Unlearning Approaches (in percent accuracy)

and making the model more robust to inference attacks. Recent works (Geva et al. 2021; Dai et al. 2022) reveal that FFN layers in Transformer blocks serve as key-value memories, storing factual knowledge. Effective unlearning requires modifying these layers, but direct updates are computationally inefficient due to their large parameter size. To address this, we integrate LoRA modules into the FFN layers, enabling low-rank, parameter-efficient updates. By making only the LoRA components trainable, our approach achieves targeted unlearning while mitigating catastrophic forgetting and maintaining model utility and efficiency

Sparse NegLoRA Recent works (Jia et al. 2024; Shah et al. 2024; Mehta et al. 2022) show that model sparsity can improve unlearning by focusing updates on a subset of parameters, reducing overfitting and computational cost. However, (Jia et al. 2024) highlights that naive gradient ascent degrades model performance as sparsity increases. To address this, we propose Sparse NegLoRA, which follows a "prune first, adapt, then ascend" approach. First, we prune a subset of model weights, then fine-tune the pruned model to restore performance. Next, Low-Rank Adaptation (LoRA) is applied for efficient updates, followed by gradient ascent to unlearn D_f. Sparse NegLoRA mitigates catastrophic forgetting while maintaining efficiency and model performance.

Experimental Setup

Our experiments aim to evaluate to evaluate the effectiveness of low-rank updates through LoRA in selectively forgetting specific data points while preserving the accuracy of the model on the remaining dataset. Furthermore, we analyze the ability of applying LoRA within sparse models, to mitigate catastrophic forgetting in gradient ascent, while ensuring effective unlearning.

Datasets

For evaluation, we used the CIFAR-10 and CIFAR-100 datasets (Krizhevsky 2009) along with a subset of the ImageNet dataset by randomly sampling 100 classes, which we refer to as ImageNet100. Check Appendix for more details.

Models

We trained a Vision Transformer (ViT) (Dosovitskiy et al. 2021) and a ResNet50 (He et al. 2015) on the aforementioned datasets. Check Appendix for more details

Baseline: We evaluate two baselines to compare against our proposed LoRA-based unlearning method:

- Retrain: Retrain the model on the retained dataset D_r
- NegGrad: Perform Gradient Ascent on the full set of model weights.
- Partial NegGrad: Gradient Ascent on a subset of the model's weights.

For Partial NegGrad in the ViT, Gradient Ascent is applied to the classifier head and the feed-forward network (FFN) and attention output projection layers in the last transformer block. For Resnet, it is applied to the convulutional and linear layers.

Sparse NegLoRA To evaluate the effectiveness of Sparse NegLoRA, we apply our method to fine-tuned sparse models at varying levels of sparsity. Specifically, we experiment with sparsity levels of 50%, 75%, 90%, and 95%, using L2 pruning. Sparse NegLoRA is then compared against the established baselines on both Vision Transformer (ViT) and ResNet architectures.

Evaluation Metrics

Our goal is to ensure that the model, after unlearning, not only removes specific data points but also retains its performance on the remaining data and remains generalizable to unseen samples. Additionally, we aim to achieve this in a parameter-efficient manner to reduce computational costs as well as memory requirements. Our metrics include:

1. Unlearning accuracy: UA = 1-Acc(D_f) is the accuracy of the unlearned model on the forget dataset. A higher UA indicates more effective unlearning of the forget dataset.
2. Retention Accuracy (RA): This is the accuracy of the unlearned model on the retain dataset, indicating how well the model maintains performance on data that should not be forgotten

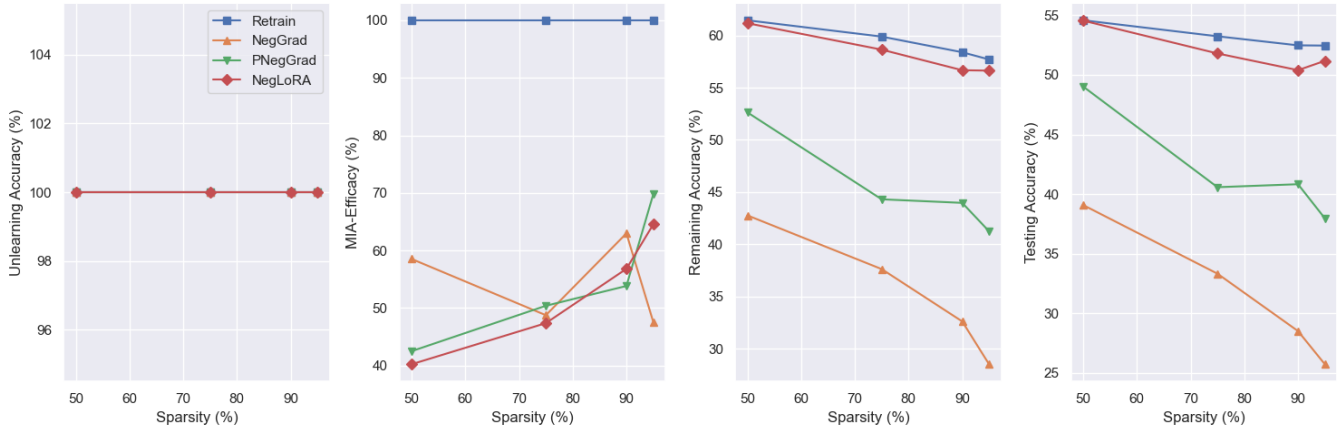

Figure 3: Sparse NegLoRA on ViT

3. Test Accuracy (TA): The accuracy of the unlearned model on a separate testing dataset containing the remaining classes, used to assess the model's generalizability after unlearning.
4. Membership Inference Attack Efficacy (MIA-Efficacy): Measures the vulnerability of the unlearned model to membership inference attacks on the forget dataset D_f using a confidence-based MIA predictor. A higher MIA-Efficacy indicates less information about D_f in the model.

Related Work

Machine Unlearning:

The problem of machine unlearning (MU) was formally introduced by (Cao and Yang 2015) in response to stricter privacy regulations, a decade of research has since followed. MU techniques can be classified into two broad categories: exact and approximate unlearning. Exact unlearning typically involves retraining the model from scratch after the removal of specific data points. (Bourtoule et al. 2020) introduced the SISA framework, which partitions data into shards and slices, with each shard serving as a weak learner that can be quickly retrained upon an unlearning request. However, the computational intensity of these methods emphasize the need for more efficient methods. (Neel, Roth, and Sharifi-Malvajerdi 2020) adopt similar DP-inspired definitions of unlearning based on the goal of indistinguishability from the retrain-from-scratch model. (Cha et al. 2024) uses utilizing adversarial examples to overcome loss of utility at the representation-level. (Tarun et al. 2024) and (Chundawat et al. 2023) learn error minimization and error maximization-based noise matrices which are used to finetune the trained model in order to do unlearning. (Thudi et al. 2022) propose a regularizer to reduce the 'verification error', which is an approximation to the distance between the unlearned model and a retrained-from-scratch model. (Shah et al. 2024) uses a Discrete Key-Value Bottleneck to perform unlearning in a model with inherent sparse representations. However, existing methods assume that at least a subset of the original training data is available at the time of unlearning. For our proposed method, we require access to no more than the unlearning data D_f , allowing seamless application for practitioners under real-world scenarios.

Results

Table 2 presents the accuracy metrics for ViT, comparing NegLoRA with the baselines. NegLoRA consistently achieves superior Remaining Accuracy (RA) and Testing Accuracy (TA) while maintaining near-perfect Unlearning Accuracy and MIA-Efficacy. Its performance remains robust egardless of the size of the dataset or the number of classes. Similarly, Appendix Table 4 demonstrates that NegLoRA outperforms baselines even on non-transformer architectures (Resnet) achieving higher RA and TA, while maintaining comparable MIA-Efficacy Furthermore, as shown in 3, applying NegLoRA to sparse models effectively mitigates catastrophic collapse. It preserves unlearning accuracy at optimal levels while delivering significantly higher Remaining Accuracy (RA) and Testing Accuracy (TA) compared to the baselines. Additionally, NegLoRA achieves MIA-Efficacy comparable to other methods, demonstrating its robustness even under high sparsity conditions.

Conclusion

This study addresses the challenge of efficient machine unlearning in light of growing privacy regulations and the need for adaptable AI systems. We introduced NegLoRA, a novel approach for efficient and effective machine unlearning using Low-Rank Adaptation (LoRA) combined with gradient ascent. Our extensive experiments across ViT and ResNet architectures demonstrate that NegLoRA consistently achieves near-perfect unlearning performace, while maintaining accuracy on the remaining classes even under varying levels of sparsity. Additionally, NegLoRA demonstrates significantly higher robustness towards inference attacks paving the road for more privacy-preserving and scalable AI systems without sacrificing utility.

References

Biderman, D.; Portes, J.; Ortiz, J. J. G.; Paul, M.; Greengard, P.; Jennings, C.; King, D.; Havens, S.; Chiley, V.; Frankle, J.; Blakeney, C.; and Cunningham, J. P. 2024. LoRA Learns Less and Forgets Less. arXiv:2405.09673.

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2020. Machine Unlearning. arXiv:1912.03817.

Cao, Y.; and Yang, J. 2015. Towards Making Systems Forget with Machine Unlearning. *2015 IEEE Symposium on Security and Privacy*, 463–480.

Cha, S.; Cho, S.; Hwang, D.; Lee, H.; Moon, T.; and Lee, M. 2024. Learning to Unlearn: Instance-wise Unlearning for Pre-trained Classifiers. arXiv:2301.11578.

Chundawat, V. S.; Tarun, A. K.; Mandal, M.; and Kankanhalli, M. 2023. Zero-Shot Machine Unlearning. *IEEE Transactions on Information Forensics and Security*, 18: 2345–2354.

Dai, D.; Dong, L.; Hao, Y.; Sui, Z.; Chang, B.; and Wei, F. 2022. Knowledge Neurons in Pretrained Transformers. arXiv:2104.08696.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929.

Frankle, J.; and Carbin, M. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. arXiv:1803.03635.

Geva, M.; Schuster, R.; Berant, J.; and Levy, O. 2021. Transformer Feed-Forward Layers Are Key-Value Memories. arXiv:2012.14913.

Golatkar, A.; Achille, A.; and Soatto, S. 2020. Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks. arXiv:1911.04933.

Graves, L.; Nagisetty, V.; and Ganesh, V. 2020. Amnesiac Machine Learning. arXiv:2010.10981.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.

Jia, J.; Liu, J.; Ram, P.; Yao, Y.; Liu, G.; Liu, Y.; Sharma, P.; and Liu, S. 2024. Model Sparsity Can Simplify Machine Unlearning. arXiv:2304.04934.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report.

Kurmanji, M.; Triantafillou, P.; Hayes, J.; and Triantafillou, E. 2023. Towards Unbounded Machine Unlearning. arXiv:2302.09880.

Mehta, R.; Pal, S.; Singh, V.; and Ravi, S. N. 2022. Deep Unlearning via Randomized Conditionally Independent Hessians. arXiv:2204.07655.

Neel, S.; Roth, A.; and Sharifi-Malvajerdi, S. 2020. Descent-to-Delete: Gradient-Based Methods for Machine Unlearning. arXiv:2007.02923.

Shah, V.; Träuble, F.; Malik, A.; Larochelle, H.; Mozer, M.; Arora, S.; Bengio, Y.; and Goyal, A. 2024. Unlearning via Sparse Representations. arXiv:2311.15268.

Tarun, A. K.; Chundawat, V. S.; Mandal, M.; and Kankanhalli, M. 2024. Fast Yet Effective Machine Unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9): 13046–13055.

Thudi, A.; Deza, G.; Chandrasekaran, V.; and Papernot, N. 2022. Unrolling SGD: Understanding Factors Influencing Machine Unlearning. arXiv:2109.13398.

Wu, K.; Zhang, J.; Peng, H.; Liu, M.; Xiao, B.; Fu, J.; and Yuan, L. 2022. TinyViT: Fast Pretraining Distillation for Small Vision Transformers. arXiv:2207.10666.

Zhang, R.; Lin, L.; Bai, Y.; and Mei, S. 2024. Negative Preference Optimization: From Catastrophic Collapse to Effective Unlearning. arXiv:2404.05868.

Appendix

Limitations and Future Scope

A promising avenue for future research is the exploration of various membership inference attack methods to further validate our hypothesis. Another potential application of this study is the extension of NegLoRA to more complex models such as Multimodal Large Language Models (MLLMs) where parameter efficiency is of the utmost importance for unlearning. While computational constraints limited our ability to explore this direction, scaling our approach to multimodal models could advance the development of adaptable, privacy-preserving AI systems capable of handling diverse and complex datasets.

Training Details and Hyperparameters

We perform all our experiments on an NVIDIA T4(x2) GPU.

Datasets

CIFAR 10 CIFAR-10 is a widely used dataset in computer vision and machine learning. It comprises 60,000 32x32 color images in 10 different classes, with 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 testing images. CIFAR-10 represents a diverse range of everyday objects, such as airplanes, automobiles, birds, and cats, making it a challenging task for image classification.

Data Augmentations: random cropping to 32x32 with 4-pixel padding, random horizontal flipping, and per-channel normalization with a mean of [0.4919, 0.4822, 0.4465] and standard deviation of [0.2023, 0.1994, 0.2010]. At test time, we resize to 32x32 and normalize.

CIFAR-100 CIFAR-100 is a complex extension of CIFAR-10, containing 100 classes with 600 images per class, split into 500 training images and 100 testing images per class. Each class is labeled with a ”fine” label. The increased number of classes make CIFAR-100 an intriguing dataset and poses a more significant challenge for models to forget specific classes while retaining knowledge of others.

Data Augmentations: random cropping to 32x32 with 4-pixel padding, 50% random horizontal flipping, and per-channel normalization with a mean of [0.5071, 0.4865, 0.4409] and standard deviation of [0.2673, 0.2564, 0.2762]. At test time, we resize to 32x32 and normalize.

ImageNet-100 ImageNet-100 is a subset of the ImageNet dataset containing 100 diverse classes, with 1300 training images and 50 validation images per class. The dataset provides a balanced mix of complexity and scale, allowing for rigorous testing of a model's ability to forget specific data points while retaining generalization across the remaining classes *Data Augmentations*: For training, we use random resized cropping (224x224) with a scale range of (0.05, 1.0), random horizontal flipping, RandAugment (n=9, m=0.5), per-channel normalization with mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225], and random erasing with a 25% probability. At test time, we resize to 224x224 and apply the same normalization.

Data Augmentations

n01968897	n01770081	n01818515	n02011460	n01496331
n01847000	n01687978	n01740131	n01537544	n01491361
n02007558	n01735189	n01630670	n01440764	n01819313
n02002556	n01667778	n01755581	n01924916	n01751748
n01984695	n01729977	n01614925	n01608432	n01443537
n01770393	n01855672	n01560419	n01592084	n01914609
n01582220	n01667114	n01985128	n01820546	n01773797
n02006656	n01986214	n01484850	n01749939	n01828970
n02018795	n01695060	n01729322	n01677366	n01734418
n01843383	n01806143	n01773549	n01775062	n01728572
n01601694	n01978287	n01930112	n01739381	n01883070
n01774384	n02037110	n01795545	n02027492	n01531178
n01944390	n01494475	n01632458	n01698640	n01675722
n01877812	n01622779	n01910747	n01860187	n01796340
n01833805	n01685808	n01756291	n01514859	n01753488
n02058221	n01632777	n01644900	n02018207	n01664065
n02028035	n02012849	n01776313	n02077923	n01774750
n01742172	n01943899	n01798484	n02051845	n01824575
n02013706	n01955084	n01773157	n01665541	n01498041
n01978455	n01693334	n01950731	n01829413	n01514668

Table 1: List of ImageNet-100 classes

Models

ViT Vision Transformer (ViT), introduced by (Dosovitskiy et al. 2021), adapts the transformer architecture to image classification by treating images as sequences of patches. We consider TinyViT from (Wu et al. 2022) with approximately 21M parameters, as it is a compact version of ViT designed to be parameter-efficient while maintaining high performance.

Resnet ResNet50 Introduced by (He et al. 2015), it facilitates the training of deep networks through residual connections, which mitigates the problem of vanishing gradients.

Detailed Evaluation Metrics

Membership Inference Attacks Membership Inference Attacks attempt to determine whether a specific data point was part of the model train data. It operates in two phases:

Config	Value
patch size	16
optimizer	AdamW
base learning rate	10^{-3}
learning rate schedule	warmup + cosine decay
weight decay	0.05
momentum	$\beta_1, \beta_2 = 0.9, 0.999$
batch size	256
warm-up epochs	20
warm-up learning rate	1×10^{-6}
training epochs	90

Table 2: ViT/16 Training Configuration Table

Config	Value
optimizer	AdamW
base learning rate	10^{-1}
learning rate schedule	cosine decay
weight decay	0.05
momentum	$\beta_1, \beta_2 = 0.9, 0.999$
batch size	128
training epochs	200
dropout rate	0.1

Table 3: Resnet50 Training Configuration Table

Config	Value
optimizer	Adam
batch size	256
LoRA rank	16
LoRA alpha	32
LoRA dropout	0.1
epochs	1/2

Table 4: LoRA Finetuning Configuration Table

- Training Phase: An MIA predictor is trained on a balanced dataset sampled from the model's remaining training set (D_r) and an external test set (D_{test}). This predictor learns to classify samples based on features such as prediction confidence.
- Testing Phase: The trained MIA predictor evaluates the forgetting dataset (D_f), aiming to classify samples as either "training" or "non-training."

The MIA-Efficacy measures the success of unlearning by quantifying how well the predictor identifies the forgetting samples as "non-training". It is formally defined as:

$$\text{MIA-Efficacy} = \frac{\text{TN}}{|D_f|}$$

where TN is the number of forgetting samples correctly classified as non-training, and $|D_f|$ is the total number of samples in D_f. A high MIA-Efficacy indicates effective unlearning.

Model	# Params	Unlearning Accuracy (UA)		MIA-Efficacy		Remaining Accuracy (RA)		Testing Accuracy (TA)		RTE
		1 Epoch	2 Epochs	1 Epoch	2 Epochs	1 Epoch	2 Epochs	1 Epoch	2 Epochs	(secs/epoch))
CIFAR100										
Retrain	23.5M	100.00	100.00	100.00	00	00	00	00	00	1.000
NegGrad	23.5M	100.00	100.00	28.60	45.40	58.05	52.91	45.55	41.47	0.025
Partial NegGrad	1.86M	100.00	100.00	58.00	48.97	60.94	45.96	47.23	37.06	0.025
NegLoRA	608K	100.00	100.00	100.00	100.00	98.97	98.66	74.19	72.21	0.017
CIFAR10										
Retrain	23.5M	100.00	0.00	100.00	100.00	38.79	39.07	38.10	34.75	1.000
NegGrad	23.5M	94.16	98.08	0.00	0.10	73.64	49.40	70.62	48.82	0.025
Partial NegGrad	20.5M	89.36	95.66	0.00	0.00	80.27	68.34	76.06	66.17	0.025
NegLoRA	20.5K	5.58	1.56	96.44	87.84	**98.17**	**95.69**	**92.92**	**90.20**	0.017

Figure 4: Results of Resnet50 When Tested Using Various Unlearning Approaches (in percent accuracy)

Run Time Efficiency We evaluate the efficiency by comparing how much faster an unlearning method is relative to retraining, i.e the run-time efficiency (RTE) of an unlearning method U is defined as:

$$\text{RTE}(U) = \frac{\text{RT}(U_R)}{\text{RT}(U)}, \quad \text{where RTE}(U) \in [0, +\infty),$$

- $\text{RT}(U)$ represents the time (in seconds) required by method U to complete unlearning.
- $\text{RT}(U_R)$ represents the time (in seconds) required for retraining the model from scratch (denoted as UR).

All methods are evaluated using the same hardware and resources for consistency. The RTE of retraining from scratch is always 1. An MU method with $\text{RTE}(U) > 1$ is faster than retraining, while a method with $\text{RTE}(U) < 1$ is slower than retraining. Ideally, the **run-time efficiency (RTE)** of a machine unlearning method should be lower than the naive approach of retraining the model from scratch.