Unlearning as an Efficient Alternative of Retraining: An Empirical Study on Forget-Set Size and Model Behavior

Joshua Sojan
University of California, Santa Cruz
josojan@ucsc.edu

Sashreek Rewatkar
University of California, Santa Cruz
srewatka@ucsc.edu

Tyler Ham
University of California, Santa Cruz
tham@ucsc.edu

Abstract

Machine unlearning, which is becoming more and more crucial for privacy and data deletion needs, attempts to eliminate the effects of training samples on models without having to retrain them from scratch. We investigate whether unlearning can function as an effective and reliable estimator of retraining on a filtered dataset. Using a Transformer trained on the AG News text classification dataset, we compare the gradient ascent unlearning to retraining across three forget-set sizes to measuring performance between a gradient ascent unlearning model and a complete retraining. We hypothesize that for small forget-set sizes, unlearning should behave similarly to retraining while being faster. As the forget-set size becomes larger, unlearning and retraining will begin to differ more in accuracy and in overall model behavior. Our results help show that Gradient Ascent unlearning can be a valid method for machine unlearning while taking note of the importance of hyperparameter tuning for a successful gradient ascent algorithm.

1 Introduction

Recent advances in deep learning have raised significant ethical challenges regarding data privacy and consent. Large-scale datasets scraped from the web are often embedded in models whose internal parameters are unknowingly trained on and memorize sensitive information. In 2015, Amazon attempted to create a model to analyze resumes, which unknowingly created a bias against women candidates [7]. Facial recognition software developed for identifying criminal suspects was revealed to have incorrectly identified suspects based on race [8]. Oftentimes, data needs to be cleaned beforehand, or models accidentally trained on said sensitive information need to be completely retrained. However, as the size of datasets and user bases of a ML model increases, the aforementioned solutions become less and less feasible [6]. *Machine unlearning* has emerged as a solution to the growing privacy concerns, enabling selective removal of data without retraining entire networks. However, most existing unlearning methods are either computationally expensive or lack formal guarantees.

In this project, we focus on gradient-ascent unlearning, where the model deliberately increases its loss on the specific samples we want it to forget. We applied Gradient Ascent based unlearning to the AG News dataset, and measured its effectiveness in comparison to completely retraining a model in both.

Preprint. Under review.

2 Methodology

Our goal is to compare gradient-ascent unlearning with retraining and compare how their behaviors differ as the forget-set size increases.

2.1 Forget-Set Construction

To study how forget-set size affects model behavior, we evaluate three removal fractions:

$$p \in \{0.05,\ 0.15,\ 0.25\}.$$

For each fraction p, we randomly sample that proportion of the training data as the *forget set*. The remaining data constitute the *retain set*.

2.2 Gradient Ascent Algorithm

To remove the influence of specific data, we apply gradient-ascent unlearning, where the loss on the selected samples is increased intentionally.

Let θ be the model's parameters and let x_f be a sample we want the model to forget. The update rule is:

$$\theta' = \theta + \beta \nabla_\theta \ell(x_f, \theta), \tag{1}$$

where $\ell(x_f, \theta)$ is the loss on the forget sample, and β is a small ascent step size. Unlike standard training, which minimizes loss, this procedure maximizes loss on the forget set, pushing the model away from representations tied to those samples. For our experiment, we chose a static ascent value of $1e^{-5}$

2.3 Model and Dataset

We decided to use the transformer model for our basis of our tests, as it provides a baseline for a complex deep learning model as the target of unlearning. We built a classification transformer using the PyTorch and PyTorch NN modules, and tested it to ensure classification accuracy.

For our dataset, we used the AG-News dataset, which balanced multi-class classification with a manageable amount of training data. The dataset contains 120,000 training samples, 7,600 test samples, and 4 classifications corresponding to different news headlines.

2.4 Experiment

Our experiment then tests the efficiency of Gradient Ascent based unlearning by creating an unlearning set of various sizes, and comparing the performance of the unlearning algorithm with a completely retraining model. We decided to test on 3 different sizes of forget sets, ranging from 5% to 30% of the total data set. For each forget set size, we obtained the accuracy metrics of retraining the retain set alongside unlearning the forget set, and compare the mean squared error from all samples.

2.5 Evaluation Metrics

We are using MSE to determine the net difference between the retraining and the unlearning. If the net difference is below a certain threshold, then this implies that the unlearning model works as an effective estimator of the retraining model.

2.6 Compute Resources

All experiments were conducted using the NVIDIA A100 GPU Hardware Accelerator available through Google Colab. The A100 provides substantially higher throughput than the standard GPUs due to its large memory bandwidth, high tensor-core density, and optimized support for mixed-precision matrix operations. Because Transformer models rely heavily on batched matrix multiplications in both self-attention layers and feed-forward blocks, the A100's architecture significantly accelerates both forward and backward passes. This reduction in per-epoch training time was important for our study, since each forget-set size requires multiple independent runs of both retraining and unlearning. Using the A100 allowed us to measure runtime differences between the two approaches with much greater precision and ensured that computational overhead did not become a limiting factor in evaluating scalability.

3 Results and Discussion

We ran three separate runs of our entire pipeline, including training a control model that trained the regular model, and running successive unlearning and retraining depending on the forget set size.

3.1 Runtime

The unlearning algorithm consistently outperformed a complete retraining in terms of total runtime

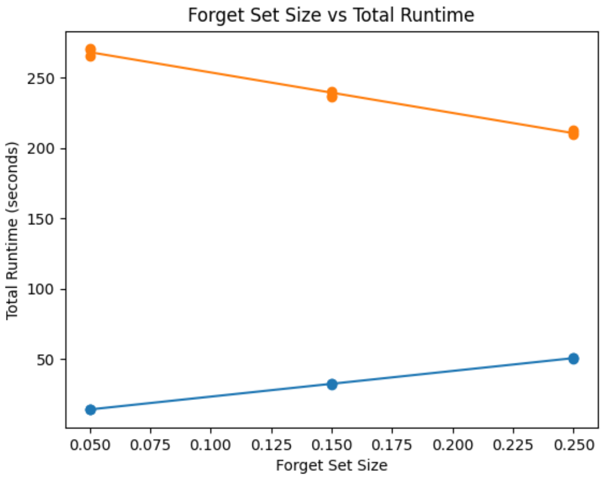


Figure 1: **Forget Set Size vs. Total Runtime.**

On a small subset of the dataset, the Gradient-Ascent unlearning algorithm was approximately 19 times faster than a complete retraining. Logically, as the size of the forget set increases (and the retain set decreases), the runtime of the unlearning and retraining model increase and decrease linearly. This is consistent with the goals of the model unlearning, as the main reason why model unlearning is preferred over complete retraining is due to the massive speed gains from only retraining a subset of the data. Figure 1 shows that the unlearning procedure is substantially more efficient at all the tested forget percentages. This figure also shows another correlation between the forget set size and the overall runtime, as its clear that the retraining runtime has a positive correlation

3.2 Accuracy

Accuracy across different ascent step sizes for Control, Unlearn, and Retrain settings show that, when the ascent step size is properly tuned (e.g., 3×10^{-7}), unlearning achieves accuracy within 1–2% of full retraining across multiple forget-set sizes. Larger step sizes (e.g., 5×10^{-6}) cause instability at higher forget set sizes, highlighting the sensitivity of the unlearning procedure to the ascent learning rate.

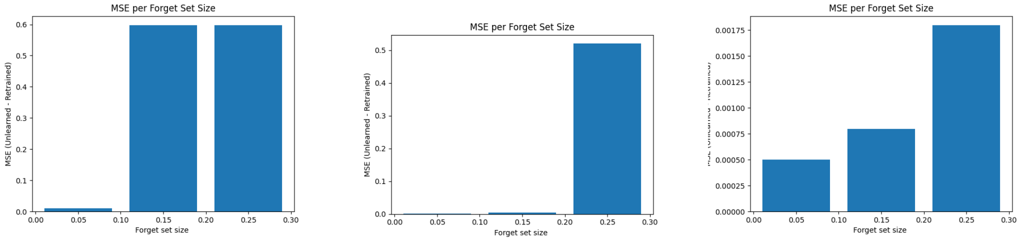
3.3 Hyperparameter Tuning

When running our tests, we noticed notable factor of gradient-ascent based unlearning - the learning rate of the unlearning played a huge factor in unlearning. As the forget set of the dataset became larger and larger, we noticed that the learning rate needed to become smaller to compensate. Doing

Table 1: Accuracy Across Different Step Sizes

	Control			**Unlearn**			**Retrain**		
	0.05	**0.15**	**0.25**	**0.05**	**0.15**	**0.25**	**0.05**	**0.15**	**0.25**
$1e^{-5}$	0.9123	-	-	0.8606	0.2530	0.2528	0.8933	0.8885	0.8938
$5e^{-6}$	0.9115	-	-	0.9091	0.8608	0.2785	0.9017	0.8903	0.8939
$3e^{-7}$	0.9161	–	–	0.8962	0.8998	0.9148	0.9152	0.9153	0.8909

further research, we found out that the likely reason for this necessary addition is that negating the loss in most neural networks is not intended behavior, meaning that without preventative measures the model will explode and the accuracy metrics for all classes will drop. This can be seen in the MSE per Forget Set Size Metric Histogram, where its clear that the learning rate meant to curb the .05 forget set set case was not small enough to curb the .15 and .25 cases. To compensate, we added gradient trimmings and a stabilization to the negation of the loss function, which did eventually make the .15 forget size fall below 1%. However, we were not able to find proper hyperparameter to tune the .25 forget set size.



(a) Ascent Step Size = $1e^{-5}$

(b) Ascent Step Size = $5e^{-6}$

(c) Ascent Step Size = $3e^{-7}$

Figure 2: Effect of Ascent Step Size on Unlearning Stability.

The unlearning algorithm reported comparable performance with suitable hyperparameter tuning, however, it also highlighted the issue of the ascent step size being too low. Although an ascent step size that is too low may cause the model to implode, a step size that is too small may fail remove to completely remove the influence of the sample.

4 Future Work

Throughout the process of this experiment, we regularly had to tune hyperparameter to match the unlearning accuracy metrics with the retraining metrics. This is of course hard to emulate in real world applications because tuning hyperparameter assumes we have a base retraining to compare to, which generally defeats the point of the unlearning. Naturally, we searched for alternative unlearning strategies that wouldn't require this specific set of data, and we eventually found the paper "Module-Aware Parameter-Efficient Machine Unlearning on Transformers", which claimed to maintain accuracy metrics between unlearning and retraining without the need to constant change the learning rate. In standard model unlearning, the model's weights are incredibly prone to exploding because gradient ascent can easily cause unrelated parameters to be directly affected by the unlearning of other parameters. This is why learning rates were not "one size fits all" - larger forget set means more parameters get affected, which means the learning needs to exponentially decrease compensate.

The strategy proposed by the paper, "Module Aware Parameter Efficient Unlearning", or MAPE Unlearning for short, aims to prevent this issue by applying learnable binary masks to each parameter within the model. These masks will "tune out" the parameters associated with the forget set, however it will retain the data associated with the retain set of the data. From here, the model can go through regular backpropagation without harming unrelated parameters. We hope to apply this approach to our unlearning model in the future, because it seems the method of surgically applying masks to the most affected parameters instead of using a sledgehammer to unlearn the features is much better at maintaining the retain set accuracy.

5 Conclusion

Our hypothesis proposed that unlearning could serve as an accurate and efficient estimator of full retraining. The results support this claim, showing that, when properly tuned, unlearning achieves accuracy within 1–2 percent of retraining while offering significant computational speedups. Several directions remain for future work in relation to our machine-unlearning experiment. As mentioned before, a strict investigation into the correlation between ascent step size and the unlearning set size would be paramount for improving the viability of machine unlearning. Additionally, some MAPE, which introduces module-aware parameter adjustments and provides stronger theoretical guarantees. Implementing MAPE or related masking-based methods would help determine whether structured parameters outperform simple gradient ascent in both stability and accuracy retention. Additionally, studying computational efficiency and the long-term stability of repeated unlearning operations across additional datasets could offer deeper insight into whether gradient-ascent unlearning is a viable alternative to full retraining in large-scale systems.

References

O. Melamed, G. Yehudai, and G. Vardi, “Provable unlearning with gradient ascent on two-layer ReLU neural networks,” *arXiv:2510.14844*, 2025.

L. Bourtoule, V. Chandrasekaran, T. Guedj, et al., “Machine unlearning,” in *Proc. IEEE Symposium on Security and Privacy*, 2021.

A. Golatkar, A. Achille, and S. Soatto, “Eternal sunshine of the spotless net: Selective forgetting in deep networks,” in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

L. Graves, V. Nagisetty, and V. Ganesh, “Amnesiac machine learning,” in *Proc. AAAI Conf. on Artificial Intelligence*, 2021.

IBM, “Exploring privacy issues in the age of AI” IBM Think Insights, 2024. [Online]. Available: `https://www.ibm.com/think/insights/ai-privacy`

Stanford Institute for Human-Centered Artificial Intelligence (HAI), “Privacy in an AI Era: How Do We Protect Our Personal Information?” 2024. [Online]. Available: `https://hai.stanford.edu/news/privacy-ai-era-how-do-we-protect-our-personal-information`

Innocence Project “When Artificial Intelligence Gets It Wrong” 2023. [Online]. Available: `https://innocenceproject.org/news/when-artificial-intelligence-gets-it-wrong/#:~:text=To%20date%2C%20six%20people%20that,match%20%E2%80%94%20all%20six%20were%20Black`

The Guardian “Amazon ditched AI recruiting tool that favored men for technical jobs” 2014. [Online]. Available: `https://www.theguardian.com/technology/2018/oct/10/amazon-hiring-ai-gender-bias-recruiting-engine`