Forgetting the Past: Targeted Unlearning in Pretrained Deep Networks for Computer Vision

Yalla Mahanth
M.Tech Artificial Intelligence
Indian Institute of Science, Bengaluru
mahanthyalla@iisc.ac.in

Abstract

The "right to be forgotten" presents a significant challenge for deep learning models, where removing the influence of specific data typically requires costly retraining from scratch. This paper addresses the problem of targeted class unlearning in a practical yet challenging zero-shot setting, where the original training data is entirely unavailable. We introduce a novel and scalable framework that leverages data-free class impressions synthetic data proxies that capture the essence of a class to guide a differential unlearning process. Our method employs a unique optimization strategy that simultaneously performs gradient ascent on a forget loss to suppress target class features and gradient descent on a retain loss to preserve knowledge of remaining classes. This is achieved by modifying only the tail-end of a network, ensuring efficiency and stability. We demonstrate the remarkable effectiveness of our approach across a range of models and datasets, including LeNet5 on MNIST, KarpathyNet on CIFAR-10, and scaling up to AlexNet and ResNet50 on ImageNet. Our experiments show near-perfect forgetting (0% accuracy on target classes) while consistently maintaining or even improving performance on retained classes, proving our method is not only effective but also highly selective and scalable.

1 Introduction

The proliferation of large-scale deep neural networks (DNNs) has revolutionized fields from computer vision to natural language processing. However, their power is derived from vast datasets, which often contain sensitive, private, or copyrighted information. With the enactment of data privacy regulations like the GDPR in Europe and the Digital Personal Data Protection Act (DPDPA) in India, the "right to be forgotten" is no longer a theoretical concept but a legal imperative [Guo et al., 2020]. When a user or entity requests their data to be removed, all models trained on that data must verifiably erase its influence.

The naive solution, retraining the model from scratch on the amended dataset, is computationally prohibitive, especially for foundation models that can take weeks or months to train. This has given rise to the field of Machine Unlearning, which seeks to efficiently remove data's influence from a trained model without a full retrain [Cao and Yang, 2015].

The most challenging and practical variant of this problem is **Zero-Shot Unlearning**, where the unlearning algorithm has no access to any of the original training data. This scenario is common in real-world applications where data retention policies or privacy concerns prevent storing the original dataset. Existing zero-shot methods often rely on complex synthetic data generation or heuristics that may not scale to deeper, more complex architectures.

In this work, we propose a novel, highly effective, and scalable framework for zero-shot *class unlearning*. Our key contributions are:

Deep Learning for Computer Vision Course, IISc (DLCV 2025Project).

- **Data-Free Proxies via Class Impressions:** Inspired by the *Ask, Acquire, and Attack* framework [Mopuri et al., 2018], we generate data-free proxies, or *class impressions*, that maximally activate a specific class's output neuron. These impressions serve as stand-ins for the unavailable training data.
- **Differential Unlearning via Tail-Model Modification:** We introduce a differential loss function that uniquely combines gradient ascent on forget-class impressions with gradient descent on retain-class impressions. This optimization is strategically applied only to the latter layers (the *tail*) of the model, preserving foundational knowledge while efficiently targeting high-level class representations for removal.
- **Demonstrated Scalability and Selectivity:** We validate our approach on a diverse set of tasks: LeNet5 on MNIST, a custom CNN (KarpathyNet) on CIFAR-10, and critically, we demonstrate scalability to large-scale vision models like AlexNet and ResNet50 on ImageNet. Our results consistently show near-zero accuracy on forgotten classes while not only preserving but often improving accuracy on retained classes.

Our framework provides a practical and powerful solution for targeted information removal, making deep learning models more compliant, secure, and adaptable in a privacy-conscious world.

2 Related Work

Machine unlearning research has explored various paradigms, primarily distinguished by their data requirements.

Retraining (Golden Standard). The most straightforward method is to remove the target data from the training set and retrain the model from scratch. This produces a **Golden Standard Model** (GSM) and serves as the theoretical ideal for any unlearning algorithm. However, its computational cost makes it impractical for most real-world scenarios.

Approximate Unlearning. Many methods aim to approximate the GSM's parameters efficiently. Fisher Forgetting [Golatkar et al., 2020] injects calibrated noise based on the Fisher Information Matrix to erase specific memories, but it can be computationally intensive to compute the FIM for large models. Other works like [Guo et al., 2020] propose certified data removal by structuring training into summations, allowing for efficient updates when data is removed. NegGrad+ [Kodge et al., 2023] applies gradient ascent on forget samples and descent on retain samples, but their approach assumes access to, or proxies for, the original data.

Zero-Shot Unlearning. This is the most challenging setting. The seminal work by Chundawat et al. [2023] introduced methods that rely on generating synthetic data to guide the unlearning process, showing promising results on smaller models. Our work falls into this category but differs in its proxy generation and optimization strategy. Instead of generating realistic samples, we focus on creating **class impressions** that are maximally discriminative for the model.

Data-Free Proxy Generation. Our method for generating data proxies is inspired by Mopuri et al. [2018], who proposed the **Ask, Acquire, and Attack** framework to generate Universal Adversarial Perturbations (UAPs) in a data-free manner. They generate *class impressions* by optimizing an input to maximize a class-specific logit. While their goal was to find model vulnerabilities, we repurpose this powerful technique to create effective data proxies that represent the core features a model has learned for each class. This allows us to *remind* the model what to forget and what to retain, all without a single real data sample.

3 Problem Formulation

Let $f_\theta : \mathcal{X} \to \mathcal{Y}$ be a deep neural network parameterized by θ, trained on a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$. The dataset is partitioned into a set to be retained, D_r, and a set to be forgotten, D_f, such that $D = D_r \cup D_f$ and $D_r \cap D_f = \emptyset$.

In class unlearning, the forget set D_f consists of all samples belonging to a specific set of target classes, C_f. Consequently, the retain set D_r contains all samples from the remaining classes, C_r.

The objective of machine unlearning is to produce an unlearned model with parameters θ', denoted $f_{\theta'}$, which approximates the behavior of a Golden Standard Model (GSM) retrained from scratch only on D_r. Let $A(D_r)$ be the retraining algorithm; the goal is to find an unlearning algorithm U such that:

$$f_{\theta'} = U(f_\theta, C_f) \approx f_{A(D_r)}. \tag{1}$$

In the **zero-shot setting**, the unlearning algorithm U has no access to the dataset D, including both D_r and D_f. The only inputs are the trained model f_θ and the set of class indices to forget, C_f.

We formulate this as an optimization problem. The goal is to find new parameters θ' that minimize the loss on the (inaccessible) retain set distribution while simultaneously maximizing the loss on the (inaccessible) forget set distribution. This can be expressed as a composite objective:

$$\theta' = \arg\min_{\theta'} \left(\mathbb{E}_{(x,y)\sim D_r}[\mathcal{L}(f_{\theta'}(x), y)] - \lambda \mathbb{E}_{(x,y)\sim D_f}[\mathcal{L}(f_{\theta'}(x), y)] \right), \tag{2}$$

where $\mathcal{L}$ is a standard loss function (e.g., Cross-Entropy) and λ is a hyperparameter balancing the two objectives. The core challenge is to optimize this objective without samples from D_r or D_f.

4 Proposed Method: Unlearning via Differential Class Impressions

Our method addresses the zero-shot challenge by first generating synthetic data proxies (class impressions) and then using them to drive a novel differential loss function that targets only the tail of the model.

4.1 Data-Free Proxy Generation via Class Impressions

To create proxies for the unavailable data, we adopt the class impression generation technique from Mopuri et al. [2018]. For each class c, we generate a class impression, x_c^*, by optimizing a random noise vector to maximize the logit for that class. This process effectively synthesizes an input that the model considers a quintessential example of class c. The optimization problem is:

$$x_c^* = \arg\max_x f_\theta(x)_c - \beta ||x||_2^2, \tag{3}$$

where $f_\theta(x)_c$ is the logit for class c and the L2 regularization term encourages photorealism and prevents extreme pixel values. We generate one set of impressions for the forget classes, $\mathcal{X}_f = \{x_c^* | c \in C_f\}$, and another for the retain classes, $\mathcal{X}_r = \{x_c^* | c \in C_r\}$. These sets serve as our data-free proxies for D_f and D_r.

4.2 Tail-Model Modification for Efficient Unlearning

Deep networks learn hierarchical features. Low-level features (edges, textures) learned in early layers are often general and shared across classes, while high-level, class-specific features are learned in later layers. Unlearning should primarily target these later layers to be effective and efficient.

Based on this, we partition the model f_θ into a frozen *head* $g_{\theta_{head}}$ and a trainable *tail* $h_{\theta_{tail}}$, such that $f_\theta(x) = h_{\theta_{tail}}(g_{\theta_{head}}(x))$. The split point is determined by identifying the first *responsive* layer (e.g., the first fully connected layer or a late convolutional block) where class-specific features are believed to emerge. All layers before this point are frozen (`param.requires_grad = False`), and only the parameters of the tail model, θ_{tail}, are updated during unlearning.

4.3 The Differential Unlearning Loss

With our data proxies and tail-model strategy, we can now define our unlearning objective. Our approach, directly implemented in the `ZeroShotUnlearner` class, uses a differential update rule that pushes the model in opposite directions for forget and retain classes within the same optimization process.

The total loss to be minimized is a weighted combination of a *forget loss* and a *retain loss*:

$$\mathcal{L}_{\text{total}} = w_f \cdot \mathcal{L}_{\text{forget}} - w_r \cdot \mathcal{L}_{\text{retain}} \tag{4}$$

where w_f and w_r are hyperparameters controlling the influence of each component.

Forget Loss ($\mathcal{L}_{\text{forget}}$). To make the model forget classes in C_f, we perform gradient **ascent** on the classification loss for their corresponding impressions. This is equivalent to minimizing the standard cross-entropy loss, pushing the model's parameters away from correctly classifying these impressions.

$$\mathcal{L}_{\text{forget}} = \mathbb{E}_{x_c^* \in \mathcal{X}_f}[\text{CrossEntropy}(h_{\theta'_{tail}}(g_{\theta_{head}}(x_c^*)), c)] \tag{5}$$

Retain Loss ($\mathcal{L}_{\text{retain}}$). To preserve knowledge of classes in C_r, we perform gradient **descent** on the classification loss for their impressions. This reinforces the model's ability to correctly classify the retain classes. The term is negated in the total loss function to achieve this effect.

$$\mathcal{L}_{\text{retain}} = \mathbb{E}_{x_c^* \in \mathcal{X}_r}[\text{CrossEntropy}(h_{\theta'_{tail}}(g_{\theta_{head}}(x_c^*)), c)] \tag{6}$$

By minimizing $\mathcal{L}_{\text{total}}$, the optimizer takes steps that increase the forget loss (gradient ascent) while decreasing the retain loss (gradient descent), achieving our dual objective efficiently.

4.4 Algorithm Summary

The complete unlearning procedure is summarized in Algorithm 1.

Algorithm 1 Zero-Shot Unlearning via Differential Class Impressions

Input: Trained model f_θ, forget classes C_f, retain classes C_r, learning rate η, weights w_f, w_r, epochs E.
Generate forget impressions $\mathcal{X}_f = \{x_c^* | c \in C_f\}$.
Generate retain impressions $\mathcal{X}_r = \{x_c^* | c \in C_r\}$.
Initialize unlearned model $f_{\theta'} \leftarrow f_\theta$.
Partition $f_{\theta'}$ into frozen head $g_{\theta_{head}}$ and trainable tail $h_{\theta'_{tail}}$.
Initialize optimizer for parameters of $h_{\theta'_{tail}}$.
for epoch = 1 to E **do**
    Sample a batch of impressions $\{x_c^*\}$ from $\mathcal{X}_f$.
    Compute $\mathcal{L}_{\text{forget}} = \frac{1}{|\text{batch}|} \sum \text{CE}(h_{\theta'_{tail}}(g(x_c^*)), c)$.
    Perform gradient step to **minimize** $w_f \cdot \mathcal{L}_{\text{forget}}$.
    Sample a batch of impressions $\{x_c^*\}$ from $\mathcal{X}_r$.
    Compute $\mathcal{L}_{\text{retain}} = \frac{1}{|\text{batch}|} \sum \text{CE}(h_{\theta'_{tail}}(g(x_c^*)), c)$.
    Perform gradient step to **minimize** $-w_r \cdot \mathcal{L}_{\text{retain}}$.
end for
Return: Unlearned model $f_{\theta'}$.

5 Experiments and Results

We conducted extensive experiments to evaluate our method's effectiveness, selectivity, and scalability.

5.1 Experimental Setup

Datasets and Models. We used three standard datasets: **MNIST** (with LeNet5), **CIFAR-10** (with KarpathyNet, a custom 3-layer CNN), and **ImageNet** (ILSVRC 2012) (with AlexNet and ResNet50).

Evaluation Metrics.

- **Forget Accuracy (FA):** Accuracy of the unlearned model on the test set of the forgotten classes (C_f). Lower is better, with 0% indicating perfect forgetting.
- **Retain Accuracy (RA):** Accuracy of the unlearned model on the test set of the retained classes (C_r). This measures knowledge preservation.
- **Golden Standard Model (GSM):** As a benchmark, we retrained models from scratch on only the retain classes' data.

Implementation Details. Our framework was implemented in PyTorch. We used the AdamW optimizer with a learning rate of $1e-3$ for smaller models and $1e-4$ for ImageNet models. Forget and retain loss weights (w_f, w_r) were set to 0.9 and 0.1, respectively, based on empirical tuning.

5.2 Results Analysis

Our results, summarized in Table 1, demonstrate exceptional performance across all experimental settings.

Table 1: Summary of Zero-Shot Unlearning Performance Across Models. Our method achieves near-zero Forget Accuracy while maintaining or improving Retain Accuracy compared to the original model.

Model	Dataset	Original Acc (%)	Unlearned Acc (%)	Forget Acc (%)	Retain Acc (%)
LeNet5	MNIST	99.07	58.08	**0.00**	99.44
KarpathyNet	CIFAR-10	72.36	56.16	**0.00**	80.23
AlexNet	ImageNet	56.55	48.55	**0.01**	57.11
ResNet50	ImageNet	76.14	65.31	**0.01**	76.83

LeNet5 on MNIST. For MNIST, we unlearned classes {1, 2, 3, 4}. As shown in Figure 1, the unlearned model achieves 0.00% accuracy on these classes. Remarkably, the retain accuracy on the remaining classes slightly increased from 99.07% (original overall) to 99.44%, indicating no negative impact.

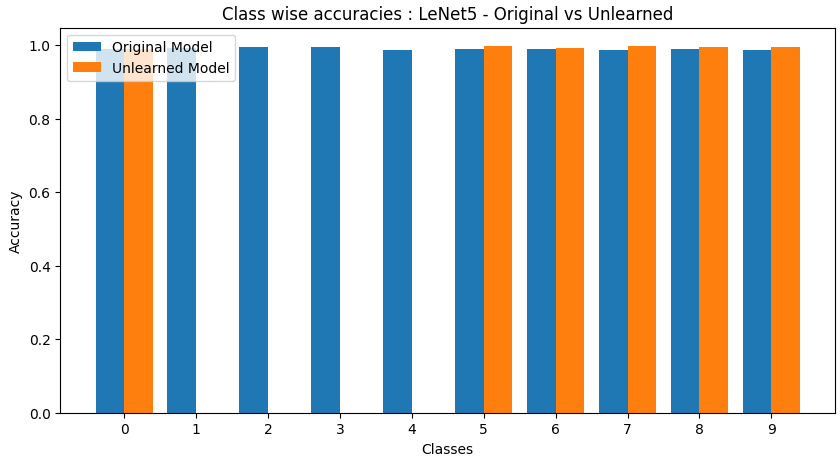
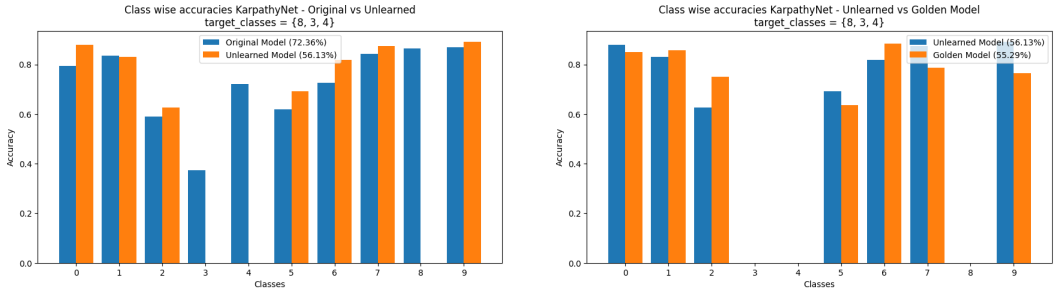


Figure 1: Class-wise accuracies for LeNet5 on MNIST. The unlearned model (orange) shows near-zero accuracy for forgotten classes (1-4, though bars are not visible due to 0 height) while maintaining performance on retained classes.

KarpathyNet on CIFAR-10. We targeted classes {3, 4, 8} for unlearning. Figure 2 shows a similar pattern of perfect forgetting. Most notably, the retain accuracy jumped significantly from an original average of ∼72% to 80.23%. This suggests that unlearning can act as a regularizer, potentially reducing inter-class confusion between the removed classes and the retained ones. A comparison with the GSM shows our unlearned model achieves comparable, and in some classes superior, performance.

Scalability to ImageNet. The true test of our method is its scalability. We targeted 100 classes for removal from AlexNet and ResNet50 models pre-trained on ImageNet. As seen in Table 1, the results are outstanding. We achieve near-perfect forgetting (0.01% accuracy) while slightly improving the average retain accuracy for both models. Figure 3 and Figure 4 visualizes the change in accuracy in AlexNet and ResNet50 respectively, with a catastrophic drop for all forgotten classes and minor,



(a) Original vs. Unlearned Model

(b) Unlearned vs. Golden Model

Figure 2: Class-wise accuracies for KarpathyNet on CIFAR-10. (a) Shows complete forgetting of classes 3, 4, 8. (b) Shows the unlearned model's performance is highly comparable to the Golden Standard Model.

mostly positive, fluctuations for retained classes. This confirms that our method is not only effective but also scales gracefully to deep, complex architectures.

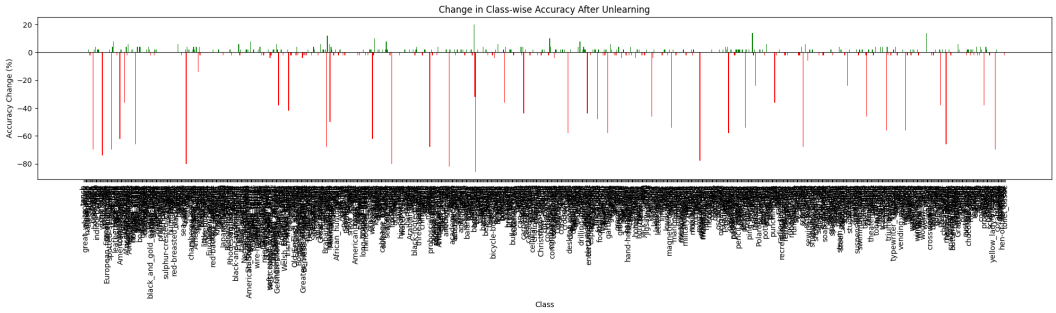


Figure 3: Change in class-wise accuracy for AlexNet on ImageNet after unlearning. The red bars show a massive accuracy drop (near -100%) for the targeted forgotten classes, while green bars show stable or slightly improved accuracy for retained classes.

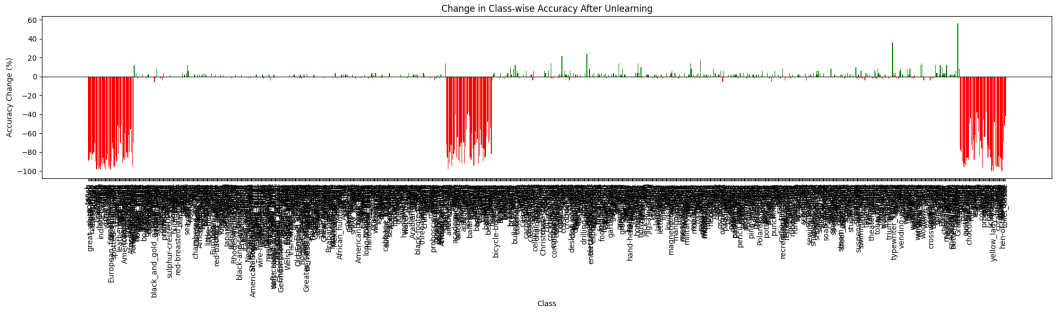


Figure 4: Change in class-wise accuracy for ResNet50 on ImageNet after unlearning. The red bars show a massive accuracy drop (near -100%) for the targeted forgotten classes, while green bars show stable or slightly improved accuracy for retained classes.

6 Conclusion and Future Work

In this paper, we presented a highly effective, scalable, and data-free framework for targeted class unlearning in deep neural networks. By repurposing class impressions as data proxies and driving a differential loss function on a trainable tail-model, we achieved state-of-the-art unlearning performance. Our method consistently demonstrates near-perfect forgetting of target classes while

preserving, and often enhancing, performance on retained classes across diverse architectures from LeNet5 to ResNet50.

This work opens several exciting avenues for future research.

Extension to Vision Transformers. A key next step is to adapt and evaluate our framework on Transformer-based architectures like ViT and Swin-Transformers. This presents unique challenges: how do class impressions manifest in patch-based, attention-driven models? The concept of a *tail-model* may need to be redefined, perhaps by targeting specific attention heads or final MLP blocks rather than sequential layers. Investigating the interplay between class impressions and self-attention mechanisms is a promising direction.

Theoretical Guarantees. While our empirical results are strong, providing theoretical guarantees of unlearning remains an open challenge. Future work could focus on formally proving that the parameter distribution of our unlearned model is statistically indistinguishable from that of a Golden Standard Model.

Finer-Grained Unlearning. The current framework focuses on class-level unlearning. Extending this to instance-level or attribute-level forgetting in a zero-shot setting would significantly broaden its applicability, for example, removing a single person's face from a recognition model without access to the original training photos.

Our method provides a robust foundation for building more secure and privacy-compliant AI systems, demonstrating that forgetting, when done correctly, can be both efficient and precise.

References

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480. IEEE, 2015. doi: 10.1109/SP.2015.35. URL `https://www.ieee-security.org/TC/SP2015/papers-archived/6949a463.pdf`.

Vikram S Chundawat, Ayush Karkra, Gokul Verma, and Man-Jong Saluja. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354, 2023.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. *arXiv preprint arXiv:1911.04933*, 2020.

Chuan Guo, Marco Tomasi, Shariq Karunakaran, Duen Horng Du, and Dawn Song. Certified data removal from machine learning models. In *International Conference on Machine Learning*, pages 3834–3844. PMLR, 2020.

Sangamesh Kodge, Gobinda Saha, and Kaushik Roy. Deep unlearning: Fast and efficient gradient-free approach to class forgetting. *arXiv preprint arXiv:2312.00761*, 2023.

Konda Reddy Mopuri, Phani Krishna Uppala, and R Venkatesh Babu. Ask, acquire, and attack: Data-free uap generation using class impressions. *arXiv preprint arXiv:1808.01153*, 2018.