UNIVERSITAT DE BARCELONA

Final Degree Project

BACHELOR'S DEGREE IN COMPUTER SCIENCE

Faculty of Mathematics and Computer Science
University of Barcelona

Challenging Forgets: Identifying and Analyzing Hard-to-Unlearn Data

Author: Sergio Gil Hernandez

Supervisor: Dr. Julio Cezar Silveira Jacques-Junior
Affiliation: Department of Mathematics and Computer Science

Barcelona, June 6, 2025

Abstract

Machine unlearning aims to remove the influence of specific data from trained models to protect privacy and comply with legal standards such as the "right to be forgotten". However, existing Machine Unlearning research has largely overlooked how the construction of Forget Sets influences unlearning success. This project addresses this gap by systematically designing and evaluating four distinct Forget Set strategies (ranging from random sampling to adversarially motivated similarity) using a ResNet-18 classifier trained on the CIFAR-10 dataset. Two unlearning techniques, basic fine-tuning and fine-tuning with final-layer perturbation, are applied. To rigorously assess performance, this study defines and applies multiple evaluation metrics: *Forgetting* (how effectively a model erases targeted data), *Utility* (how well it retains performance on retained data), and a composite metric that balances both. The results reveal how Forget Set composition critically affects the effectiveness of Machine Unlearning strategies, offering new insights for future research and development.

Resumen

El “Machine Unlearning” tiene como objetivo eliminar la influencia de datos específicos en modelos ya entrenados, con el fin de proteger la privacidad y cumplir con normativas legales como el “derecho al olvido”. Sin embargo, gran parte de la investigación existente ha pasado por alto cómo la composición de los *Forget Sets* influye en el éxito del proceso de olvido. Este proyecto aborda dicha carencia mediante el diseño y evaluación sistemática de cuatro estrategias distintas de selección de *Forget Sets*, que van desde muestreo aleatorio hasta enfoques motivados por similitud estructural adversaria, utilizando un clasificador ResNet-18 entrenado sobre el conjunto de datos CIFAR-10. Se aplican dos técnicas de desaprendizaje: fine-tuning básico y fine-tuning con reinicialización de la capa final. Para evaluar rigurosamente el rendimiento, se definen y aplican diversas métricas: *Forgetting* (grado en que el modelo olvida los datos objetivo), *Utility* (capacidad para mantener el rendimiento sobre los datos retenidos) y una métrica compuesta que equilibra ambas. Los resultados demuestran que la composición del *Forget Set* afecta críticamente la eficacia de las estrategias de *Machine Unlearning*, proporcionando nuevas perspectivas para la investigación futura.

Resum

El "Machine Unlearning" té com a objectiu eliminar la influència de dades específiques en models ja entrenats per tal de protegir la privacitat i complir amb normatives legals com el "dret a l'oblit". Tanmateix, gran part de la recerca existent ha passat per alt com la composició dels *Forget Sets* influeix en l'èxit del procés d'oblit. Aquest projecte aborda aquesta mancança mitjançant el disseny i l'avaluació sistemàtica de quatre estratègies diferents de selecció de *Forget Sets*, que van des de mostres aleatòries fins a aproximacions motivades per la similitud estructural de tipus adversari, utilitzant un classificador ResNet-18 entrenat sobre el conjunt de dades CIFAR-10. S'apliquen dues tècniques de desaprendre: fine-tuning bàsic i fine-tuning amb reinicialització de la capa final. Per avaluar el rendiment de forma rigorosa, s'han definit i aplicat diverses mètriques: *Forgetting* (grau amb què el model oblida les dades objectiu), *Utility* (capacitat per mantenir el rendiment sobre les dades retingudes), i una mètrica composta que equilibra ambdues. Els resultats mostren que la composició del *Forget Set* afecta críticament l'eficàcia de les estratègies de *Machine Unlearning*, aportant noves perspectives per a la recerca futura.

Acknowledgments

I am sincerely thankful to my advisor, Julio Cezar Silveira Jacques-Junior, for his unwavering guidance, motivation, and dedication throughout this project. His invaluable help, the generous time he devoted, and his continuous support played a key role in making this work possible.

I would also like to extend my heartfelt thanks to my family, whose constant encouragement, patience, and belief in me have been a source of strength during every stage of this process.

Contents

1. Introduction

1.1. Machine Unlearning: Definition and Importance

The concept of Machine Unlearning (MU) emerges due to the increasing need to protect user privacy and comply with regulations such as the "right to be forgotten" [1]. It refers to a set of techniques designed to reduce the influence of specific data on a machine learning model that has already been trained. This process is crucial when it is necessary to remove irrelevant, outdated or sensitive information, ensuring that the model no longer reflects certain data.

Formally, let M be a model trained on a dataset D using a training algorithm A, where we do not distinguish between M and its parameters, and write $M = A(D)$. An unlearning query is typically identified by a Forget Set D_f and a Retain Set $D_r = D \setminus D_f$. The goal of an unlearning algorithm U is to remove from M the influence of D_f, producing an unlearned model $M_u = U(M, D_f, D_r)$ [2]. A generic Machine Unlearning (MU) pipeline is illustrated in Figure 1.

There are two main approaches to Machine Unlearning: exact unlearning, which thoroughly eliminates data influence, and approximate unlearning, which aims to efficiently reduce the data impact while maintaining model performance. The field is gaining traction due to its importance in both privacy protection and data governance, with challenges remaining in terms of verifying the effectiveness of unlearning and ensuring efficiency [3, 4]. A more detailed discussion of these two main approaches, including their strengths and limitations, is provided in the Related Work section (Section 2).

1.2. Motivations

A significant limitation in current Machine Unlearning research lies in the insufficient exploration of strategies for constructing Forget Sets. Most existing approaches prioritize the development of efficient unlearning mechanisms, but pay little attention to the selection criteria for the data points to be forgotten. In many cases, Forget Sets are chosen arbitrarily, randomly or based on loosely defined heuristics without a thorough examination of how these choices impact the effectiveness of the unlearning process. This oversight raises an important question: can the composition of the Forget Set influence the overall success of unlearning?

This work is motivated by recent studies in the field [5], which highlight that the strat-

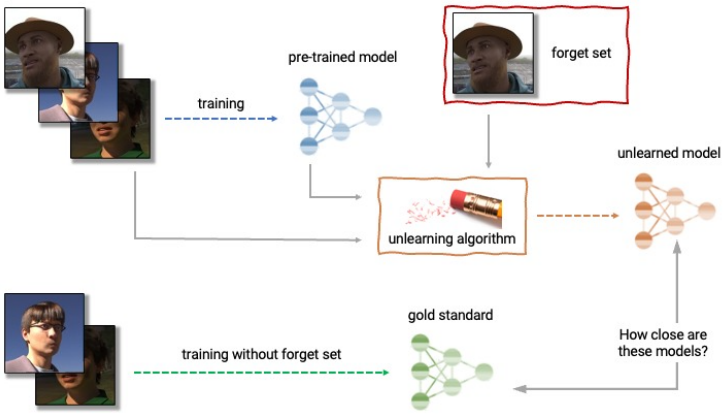

Figure 1: Generic Machine Unlearning (MU) pipeline. A model M is initially trained on a dataset D. Upon receiving a forget request defined by a subset D_f, an unlearning algorithm U removes the influence of D_f, resulting in an unlearned model M_u that approximates the behavior of a reference model trained from scratch on the retained set D_r. Taken from the diagram presented in Google's Machine Unlearning Challenge announcement `https://research.google/blog/announcing-the-first-machine-unlearning-challenge/`.

egy used to select the Forget Set plays a crucial role in both the efficiency and efficacy of unlearning. Different unlearning strategies can lead to vastly different outcomes, affecting not only how well a model forgets specific information, but also how much unintended degradation occurs in its overall performance. If the Forget Set is poorly chosen, the model may retain traces of the forgotten information or suffer an unnecessary loss of accuracy. Conversely, a well-constructed Forget Set could encourage the research community to further explore the potential for more powerful and efficient Machine Unlearning models, while optimizing computational resources and ensuring a thorough removal of unwanted data influence.

Moreover, despite recent advances, there is still no clear consensus or formal guarantee regarding the forgetting capabilities of approximate unlearning techniques. As a result, it is common practice to assess their effectiveness by comparing their outcomes to those of exact unlearning methods [6, 7]. This comparative approach underscores the need for a deeper understanding of how Forget Set composition interacts with different unlearning paradigms, and highlights the potential benefits of more principled selection strategies.

1.3. Objectives

1.3.1 General Objective

The general purpose of this study is to evaluate different Machine Unlearning strategies using multiple and carefully designed Forget Sets. By systematically examining different approaches, this research aims to identify optimal and suboptimal scenarios, assess

the effects of diverse Forget Sets and unlearning strategies, and establish a foundational understanding of their broader implications. Additionally, it seeks to determine which types of data are more or less difficult to forget and how to achieve the best performance in terms of Forgetting and Utility.

Machine unlearning can be applied to a wide range of scenarios and model architectures, from large-scale language models to personalized recommendation systems and medical applications. However, given the computational demands associated with training and evaluating large models, this work adopts a more constrained yet representative setting as a proof of concept. Specifically, we define image classification as a case study and use a simple, well-known dataset to conduct controlled experiments. This setup enables a focused and feasible evaluation of unlearning behaviors, while still reflecting meaningful dynamics that can generalize to more complex scenarios.

To this end, we apply the selected unlearning methods to a previously trained ResNet-18 model on the CIFAR-10[1] classification dataset, and assess their capacity to remove the desired information while preserving satisfactory predictive performance. Further details about the dataset and experimental design are provided in Section 3 and Section 4, respectively.

1.3.2 Specific Objectives

Specifically, this project aims to:

- Investigate and implement various strategies for constructing Forget Sets, systematically comparing how different unlearning methods perform with each set to identify the best and worst-case scenarios.
- Implement distinct unlearning techniques with the goal of retaining model accuracy and achieving effective forgetting (within the context of each Forget Set), in order to facilitate a thorough comparative analysis.
- Define precise and interpretable evaluation metrics to rigorously assess model performance in terms of both forgetting efficacy and retention of useful knowledge.
- Analyze and compare all obtained results to derive insights into the impact of different Forget Set selection strategies and unlearning methods, ultimately contributing to a deeper understanding of optimal approaches for effective and reliable Machine Unlearning.
- Use a ResNet-18 model trained on the CIFAR-10 dataset as a use case to apply the unlearning method, serving as a proof of concept to evaluate its capacity to eliminate the targeted information while preserving satisfactory predictive performance.

1.4. Structure of the Thesis

The project is divided into the following sections:

[1]CIFAR-10 dataset: `https://www.cs.toronto.edu/~kriz/cifar.html`

- **Related Work:** Section 2 reviews previous research on Machine Unlearning, unlearning methods and Forget Set selection strategies. It highlights key findings, methodologies, and gaps in the literature where this project can contribute.
- **Proposed Approach:** Section 3 describes the different unlearning models, techniques and evaluation metrics implemented in the study. It details how Forget Sets were constructed, the algorithms used, and any modifications or improvements made to existing methods.
- **Experiments and Results:** Section 4 presents the experiments conducted to evaluate the performance of various unlearning strategies. It includes a comparison of different Forget Sets, outlines the initial hypotheses, reports the results for each metric, and discusses the key findings.
- **Conclusions and Future Work:** Section 5 summarizes the main insights gained from the study, discussing the effectiveness of different unlearning approaches and Forget Set designs. It also outlines limitations and proposes future research directions to improve unlearning methods and evaluation strategies.

2. Related Work

This section reviews the foundational concepts and recent developments in the field of Machine Unlearning. We examine the main categories of MU techniques, highlighting their respective advantages, limitations, and areas of application. The aim is to provide the reader with a comprehensive understanding of how different approaches operate and the trade-offs they involve in terms of forgetting effectiveness, computational efficiency, and model Utility.

The structure of this section is as follows: First, we present and categorize existing MU methods, distinguishing between exact and approximate unlearning techniques. We then discuss commonly used evaluation metrics that allow researchers to assess the effectiveness and reliability of unlearning mechanisms. Finally, we explore recent work that focuses on the notion of challenging forgets, which emphasizes the importance of the composition of Forget Sets and the difficulty of forgetting certain types of data. This organization aims to contextualize the motivation for our study and highlight the gaps this work seeks to address.

2.1. Naive Retraining

Naive retraining is the most direct and basic method for Machine Unlearning. In this approach, the model is entirely retrained from scratch using a modified dataset that excludes the data to be unlearned. This ensures that the reference model has no residual influence from the removed data. In general terms, naive retraining is considered the baseline for unlearning techniques because it guarantees the complete and accurate removal of undesired information [3].

2.1.1 Limitations

Although naive retraining provides the strongest guarantees for unlearning, it is often impractical in real-world applications because it presents several limitations:

1. **Computational Expense:** Retraining a model from scratch, especially when dealing with complex architectures, demands significant computational resources and time. For large-scale datasets, the cost can quickly become prohibitive.

2. **Inaccessibility of Data:** In scenarios like federated learning [8] or distributed systems, the original training dataset might no longer be available after the initial train-

ing. Without access to the complete dataset, retraining is not feasible as it depends entirely on it.

3. **Lack of Scalability:** As datasets grow and models become more complex, naive retraining becomes increasingly difficult to scale, even more when frequent unlearning requests are made.

Despite these challenges, naive retraining remains the benchmark against which other unlearning techniques are evaluated due to its ability to ensure complete removal of the target data's influence.

2.2. SISA

SISA (Sharded, Isolated, Sliced, and Aggregated) is an exact unlearning framework that enhances the efficiency and scalability of the unlearning process by structurally limiting the influence of individual data points during training. It achieves this by partitioning and incrementally processing the dataset, making unlearning requests localized and computationally lightweight. SISA is particularly efficient even under worst-case scenarios, where unlearning requests are uniformly distributed across the training set [9]. The general pipeline of SISA, discussed next, is illustrated in Figure 2.

2.2.1 SISA General Process

SISA operates through four coordinated steps:

- **Sharding:** The dataset is split into multiple independent shards, each containing a disjoint subset of the training data. Models are trained separately per shard, allowing unlearning operations to be confined to the affected shard, significantly reducing retraining costs.
- **Isolation:** Within each shard, data is organized to minimize cross-sample influence. This isolation enhances the traceability of individual data points and ensures that forgetting a sample does not inadvertently affect unrelated data.
- **Slicing and Incremental Learning:** Each shard is further divided into sequential slices. Training proceeds incrementally over these slices, and the model state is saved after each slice. This enables efficient unlearning by allowing retraining to resume from the slice where the forgotten sample was introduced, rather than from the beginning.
- **Aggregation:** After all shards and slices have been trained, their models are aggregated into a single global model. This step consolidates the contributions of each isolated unit while preserving the efficiency benefits of distributed training.

Parameter Archiving: A key aspect of SISA is the archiving of model parameters after each training slice. These checkpoints are critical to supporting fast and localized retraining during unlearning operations, further reducing computational overhead.

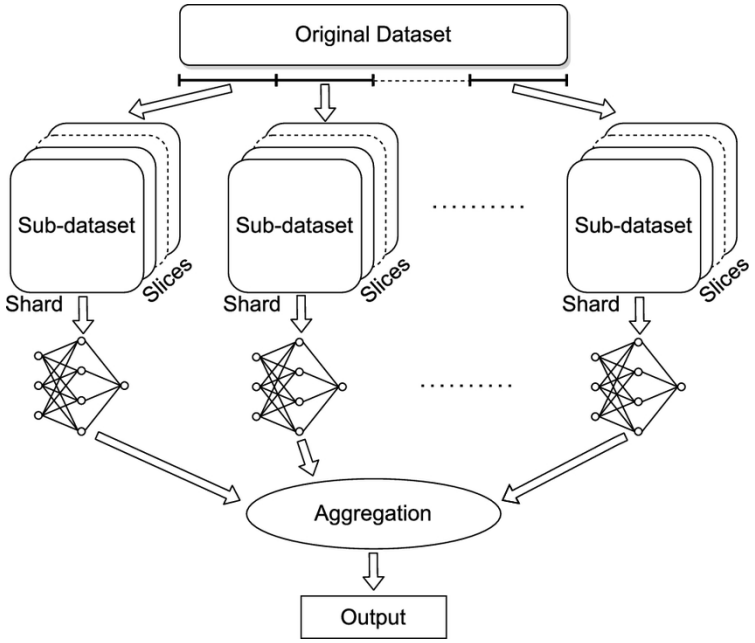

Figure 2: Overview of the SISA framework, obtained from [10]. The original dataset is divided into several independent shards (sub-datasets), each of which is further partitioned into sequential slices. Models are trained incrementally on each slice within a shard. After training is completed for all shards and slices, the resulting models are aggregated to produce the final model output. This design enables efficient and localized unlearning by retraining only the affected shard and slice when a data removal request is made.

Unlearning with SISA: Upon receiving an unlearning request, SISA identifies the specific shard and slice containing the target data point. Retraining resumes from the archived model state immediately preceding that slice, ensuring that the data point's influence is removed without affecting the rest of the model.

2.2.2 Limitations of SISA

Despite its efficiency advantages, SISA presents several limitations that affect its scalability and accuracy, as detailed next.

First, the *sharding* mechanism can degrade model performance as the number of shards increases. This is particularly problematic in tasks with high class imbalance or complex patterns, such as image classification, where reducing the number of samples per shard can harm generalization [11].

Second, the use of *slicing* introduces a linear storage overhead, as intermediate model checkpoints must be archived for each slice. This overhead becomes increasingly burdensome with large models or when a high number of slices is used.

Finally, while sharding and slicing accelerate unlearning for sparse requests, their effectiveness diminishes as request volume grows. When the number of unlearning operations crosses a certain threshold, SISA's performance begins to approach that of traditional

retraining methods. In large-scale datasets like ImageNet, this degradation is more pronounced, and the accuracy of SISA-trained models may fall behind simpler baselines like batch K-nearest neighbors (K-NN) [11].

These trade-offs underscore that while SISA is well-suited to scenarios involving occasional, localized unlearning requests, its benefits may not extend to all domains or workloads—especially those requiring high model fidelity or frequent forgetting.

2.3. Fine-Tuning

Fine-Tuning is one of the most straightforward and widely adopted techniques for adapting a pre-trained machine learning model to a new task or dataset. In the context of Machine Unlearning, Fine-Tuning serves as a *baseline method* for removing specific information from a model. The core idea is to retrain the model on a modified version of the original dataset, where the Forget Set has been excluded, in order to reduce the model's dependence on that data.

This process begins with a model that has been trained on a large and comprehensive dataset, allowing it to learn general representations. During Fine-Tuning, this pre-trained model is then updated (typically through backpropagation) using a new training set composed only of the retained data. The learning rate is often reduced in this phase to preserve previously acquired knowledge while allowing the model to adjust to the updated data distribution.

Importantly, Fine-Tuning leverages the phenomenon of *catastrophic forgetting* — a well-documented behavior in continual learning where a model rapidly loses previously acquired knowledge when trained on new data. In the context of MU, this behavior is intentionally exploited: by excluding the Forget Set and retraining on the retained samples, the model is nudged to overwrite representations associated with the removed data. This makes Fine-Tuning an effective, though approximate, strategy for inducing forgetting. As discussed by [6], catastrophic forgetting forms the foundation of several unlearning approaches aimed at eliminating specific information without full retraining.

However, Fine-Tuning does not offer formal guarantees of complete forgetting. Residual traces of the forgotten data may persist in the model's internal parameters, particularly when the Forget Set had a significant or unique impact on the learned features.

In summary, Fine-Tuning is a conceptually simple and accessible approach for unlearning. It provides a flexible way to update a model and serves as a useful baseline for evaluating more advanced methods that incorporate explicit forgetting mechanisms or formal guarantees.

2.3.1 Limitations of Fine-Tuning

While Fine-Tuning can be effective in many scenarios, it presents several limitations when used for unlearning:

- **Residual Traces of Forgotten Data:** Even though Fine-Tuning aims to remove the influence of specific data, it does not guarantee complete erasure. The model may retain residual traces of the forgotten data, particularly if that data had a strong

impact on the learned representations. This is especially problematic in privacy-sensitive scenarios, where the goal is to ensure that the forgotten data cannot be reconstructed or recalled by the model during inference.

- **Risk of Overfitting:** Fine-Tuning on a reduced dataset increases the likelihood of overfitting, especially if the remaining data is insufficient or unbalanced. Overfitting occurs when the model becomes too tailored to the new training set, losing its ability to generalize to unseen data. In the context of unlearning, this can result in degraded performance on unrelated tasks or test samples.

These limitations underscore the need for more refined and robust methods for Machine Unlearning, such as Fine-Tuning with Weight Perturbation, which aims to address some of the issues related to residual traces and overfitting.

2.4. Fine-Tuning with Weight Perturbation

Fine-Tuning with Weight Perturbation is a variant of Fine-Tuning that aims to unlearn specific data points by introducing small, targeted perturbations to the model's weights, rather than retraining the entire model from scratch. This approach involves adding perturbations to the model before performing fine-tuning on the already trained parameters. The goal is to allow the model to recover from the influence of previous data while trying to reduce its impact. One challenge with this method is identifying which parameters should be perturbed, as well as defining the optimal magnitude of the perturbation, which is crucial for achieving effective unlearning without significantly affecting the model's overall performance [12]. Despite these challenges, the approach offers significant computational efficiency, particularly in scenarios where retraining the model from scratch would be too resource-intensive.

2.4.1 Key Mechanisms and Workflow

The core of Fine-Tuning with Weight Perturbation lies in identifying and perturbing parameters that are most relevant to the data to be unlearned. This targeted adjustment aims to suppress the influence of the undesired data while preserving the model's generalization ability across the remaining dataset.

1. **Identification of Critical Parameters:** To selectively unlearn specific data points, the first step involves identifying which parameters of the model are most influenced by the data in question. This is typically done through gradient-based sensitivity analysis. Specifically, the gradient of the loss function with respect to each parameter is computed for the data to be forgotten. Parameters with the highest gradient magnitudes are considered most critical, as they have the strongest influence on the model's predictions for the target data [12].

2. **Perturbation Strategies:** After identifying the critical parameters, small, controlled perturbations are applied to reduce their sensitivity to the forget data. According to [12], two primary strategies have been explored:

- *Top-K Perturbation:* The top K parameters with the highest sensitivity are selected and perturbed. This strategy focuses the adjustment on the most influential components of the model.
- *Random-K Perturbation:* A random subset of K parameters is perturbed. While less precise, the added randomness can still interfere with the encoded information related to the forget data and promote unlearning.
- *Alternative Perturbation Approaches:* Other forms of perturbation can also be considered, such as reinitializing the weights of a specific layer or group of layers before fine-tuning. For example, [13] propose resetting the weights of the final classifier layer as a way to erase specific task-related knowledge prior to retraining. These broader strategies highlight the importance and difficulty of defining an effective perturbation scheme that ensures forgetting while maintaining overall model Utility.

2.4.2 Advantages and Limitations

Advantages:

- **Computational Efficiency:** By focusing only on a subset of parameters, the approach greatly reduces the cost compared to full retraining, making it attractive in resource-constrained scenarios.
- **Targeted Forgetting:** Perturbing high-sensitivity parameters allows for more direct removal of data influence, potentially improving unlearning effectiveness without requiring many epochs of retraining.
- **Lower Risk of Overfitting:** Since fewer parameters are adjusted, the model is less likely to overfit to the reduced dataset, preserving its ability to generalize to unseen data.

Limitations:

- **Sensitivity to Perturbation Choice:** The success of the method depends critically on how well the influential parameters and the perturbation values are selected. Poor choices may result in insufficient forgetting or degraded model accuracy.
- **Limited Guarantees:** Unlike some exact unlearning methods, this approach provides no formal guarantees that the target data has been completely forgotten.
- **Lack of Universality:** The method may not generalize well to all model architectures or unlearning scenarios, requiring manual tuning or hybrid strategies to achieve satisfactory results.

2.5. Common Evaluation Metrics in Machine Unlearning

Evaluating the effectiveness of Machine Unlearning methods is a crucial aspect of research in this field. The goal is not only to ensure that a model forgets the designated

data, but also to verify that it continues to perform well on the remaining tasks. A variety of metrics have been proposed in the literature to measure both the forgetting success and the retention of Utility. This section introduces the most common evaluation metrics used in MU, which will be important to contextualize the analysis presented in later sections.

One of the most frequently used metrics is *Membership Inference Accuracy (MIA)*, which assesses whether an adversary can infer if a specific sample was part of the training data. After a successful unlearning process, the MIA score for the forgotten samples should drop to random chance levels. Another common approach is to directly measure the *forgetting accuracy*, that is, the classification accuracy of the model on the Forget Set after unlearning. Ideally, this accuracy should decrease significantly, indicating that the model no longer retains useful information about those samples.

In contrast, *retention accuracy* evaluates how well the model performs on the remaining (non-forgotten) data. A robust unlearning technique should maintain high retention accuracy to ensure that only the target data is affected. The trade-off between forgetting and retention is central to evaluating MU methods: a good method maximizes forgetting while minimizing Utility loss.

A popular baseline used in many studies is to compare the unlearned model to a reference model retrained from scratch without the forget data. This leads to the definition of *approximation-based metrics*, such as the distance between model parameters, differences in output distributions (e.g., using KL-divergence), or prediction consistency across datasets. The closer the unlearned model is to this ideal reference model, the more effective the unlearning.

Finally, more advanced and resource-intensive evaluation strategies have emerged. For example, Google's work on SISA [11] and later efforts by others [14] propose comprehensive auditing frameworks that rely on techniques such as shadow models, influence functions, or privacy risk estimators. These methods offer a more rigorous assessment of residual data influence but require significant computational resources, making them less practical for smaller-scale or exploratory studies.

In this work, we adapt a selection of these standard metrics and other metrics developed by us as a practical and interpretable way to analyze unlearning outcomes. Further details on the metrics used in our experiments are presented in Section 4.

2.6. Finding the Worst and Easiest Scenarios

This topic was first addressed by the paper *"Challenging Forgets: Unveiling the Worst-Case Forget Sets in Machine Unlearning"* [5], which serves as the main motivation for this project. The study highlights that evaluating Machine Unlearning methods using randomly selected Forget Sets may overlook important insights about the robustness and limitations of these approaches. In practice, certain data points are inherently more difficult to forget than others, and assessing the performance of unlearning strategies under these more challenging conditions is essential to better understand their behavior and improve their effectiveness.

The authors propose an adversarial evaluation perspective by introducing the concept of *worst-case Forget Sets*—subsets of training data that are particularly hard to erase from a

model. Rather than relying on arbitrary or random selection, their goal is to systematically identify those samples whose removal is most difficult for a given unlearning algorithm, typically leading to poor forgetting effectiveness or significant Utility degradation.

To do so, the paper introduces a bi-level optimization framework, structured as follows:

- **Upper level:** This component searches for the subset of data points whose removal will maximize the difficulty of the unlearning process. More specifically, it selects the Forget Set that results in the largest discrepancy between the output of the unlearned model and a fully reference model (the "oracle"). This stage effectively simulates an adversary aiming to expose the weaknesses of the unlearning method.

- **Lower level:** At this level, the selected Forget Set is processed through the chosen unlearning algorithm. The model is updated (e.g., via approximate retraining or fine-tuning), and its performance is evaluated using metrics that capture both forgetting success (e.g., influence scores, prediction changes) and Utility preservation (e.g., accuracy on the retained data). This step provides feedback to guide the upper-level search.

This iterative process allows the discovery of data samples that are particularly resistant to forgetting. The study shows that these worst-case Forget Sets often include samples with high influence on model parameters, such as those located near decision boundaries, from rare classes, or representing atypical patterns. Forgetting such data is more likely to leave residual traces in the model or harm its generalization ability.

On the other hand, identifying the easiest scenarios (samples that are simple to forget without impacting model Utility) can also be informative. These may include redundant or less informative samples that contribute little to the overall model behavior.

2.6.1 Main findings and open opportunities

The authors conducted extensive experiments using diverse datasets, including CIFAR-10, CIFAR-100, CelebA, Tiny ImageNet, and ImageNet, to evaluate the performance of different Machine Unlearning methods. These datasets cover a range of complexity, from simple object categories to high-resolution images, allowing for a comprehensive analysis. They also tested various models, such as image classifiers (e.g., ResNet and Vision Transformers) and generative models (e.g., VAEs and GANs), to assess how well unlearning methods remove specific data while preserving overall performance.

The results revealed significant differences in the effectiveness of unlearning methods when applied to worst-case Forget Sets compared to randomly selected subsets. While traditional evaluations based on random data removal suggested that many unlearning techniques performed well, the worst-case Forget Sets exposed their limitations. Some data points proved much harder to forget, revealing that certain unlearning approaches fail to fully erase sensitive information.

Among the evaluated methods, exact unlearning (which requires retraining the model entirely) was the most effective, although computationally expensive. Approximate unlearning approaches, such as fine-tuning and weight perturbation, struggled more with worst-case Forget Sets, often leaving residual traces of the forgotten data. Gradient-based

unlearning showed some promise but remained unreliable when tested on adversarially selected subsets.

These findings emphasize the need to assess unlearning techniques in adversarial conditions rather than relying on random removal. By identifying the hardest-to-forget data points, the study highlights weaknesses in current approaches and underscores the necessity for more robust unlearning methods that can handle real-world challenges effectively.

Motivated by these insights, this work proposes an alternative approach to define and approximate worst-case scenarios in a computationally efficient manner. Our method focuses on simplicity and scalability, making it suitable for practical use in a variety of settings. While it may not yield optimally adversarial Forget Sets like bi-level optimization methods, it provides a strong approximation that can help reveal model weaknesses with far less computational overhead. Given that this is a relatively new research direction in Machine Unlearning, our goal is to advance the exploration of how Forget Set composition influences unlearning effectiveness and to contribute new insights for designing more resilient Machine Unlearning techniques.

3. Proposed Approach

To achieve the objectives outlined in this study, it is necessary to implement and utilize multiple machine learning approaches. These methods provide the foundation for understanding the extent to which Machine Unlearning can be effectively achieved and the trade-offs involved in the process.

In this section, we describe the models and techniques used to conduct our experiments. We establish a baseline for comparison, introduce an alternative approach that represents an ideal forgetting scenario, and explore different unlearning strategies designed to selectively remove information while preserving overall model performance.

This section also outlines the strategies employed to construct the various Forget Sets, each with a distinct motivation aimed at identifying challenging and representative scenarios for evaluating unlearning performance. Finally, we present the evaluation metrics adopted to assess both forgetting effectiveness and the preservation of Utility.

3.1. Original Model

The Original Model (M_o) serves as the starting point for the unlearning process, representing a fully trained classifier on the CIFAR-10 dataset before any data removal.

For this study, we selected ResNet-18 [15] as the architecture for the classifier. ResNet-18 is a widely used convolutional neural network (CNN) known for its efficiency and strong performance on image classification tasks. Its residual connections help mitigate vanishing gradient issues, allowing for deeper networks while maintaining stability during training. Given the relatively small size and complexity of CIFAR-10, ResNet-18 strikes a balance between computational efficiency and accuracy, making it an ideal choice for our experiments.

Moreover, training a model from scratch was neither feasible due to resource limitations nor the primary objective of this study. Instead, we utilized a pretrained ResNet-18 model, which served as a reliable and consistent baseline for evaluating various unlearning strategies. This model was originally trained on the CIFAR-10 dataset, which consists of 50,000 training images across 10 classes. The training process spanned 200 epochs, ensuring a well-converged model with strong generalization performance. The pretrained weights are publicly available[2], allowing reproducibility and consistency in experimentation.

[2] `https://storage.googleapis.com/unlearning-challenge/weights_resnet18_cifar10.pth`

It is important to clarify that the samples selected for unlearning were drawn exclusively from the CIFAR-10 training set. This guarantees that all forget samples were part of the original training data used for the pretrained ResNet-18 model. As CIFAR-10 has a standardized split, we avoid any overlap or confusion between training, validation and test sets. Consequently, we ensure that unlearning operations target data that indeed contributed to the original training process, making the evaluation both meaningful and reliable.

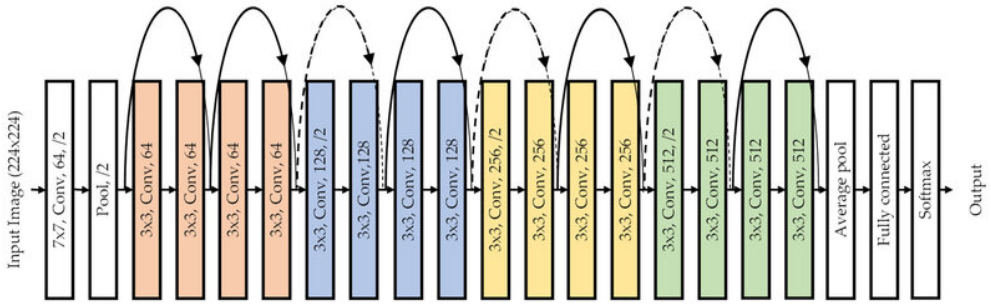

Figure 3: Architecture of the ResNet-18 model used in our experiments. ResNet-18 is a convolutional neural network composed of 18 layers, including convolutional layers, batch normalization, ReLU activations, and shortcut connections that enable residual learning. This architecture is widely used for image classification tasks due to its balance between performance and computational efficiency. In our setup, feature representations are extracted from the penultimate layer to compute similarity between samples for unlearning analysis.

3.2. Reference Model

The reference model (M_r) represents the standard benchmark for evaluating the effectiveness of Machine Unlearning methods. Rather than adapting an already trained model, this approach involves training a new model from scratch after removing the Forget Set. As a result, M_r has never been exposed to the data that is intended to be forgotten, making it an ideal reference point to measure how closely an unlearning method approximates the outcome of full retraining. This naive retraining baseline is widely used in the literature to assess the success of unlearning strategies in fully eliminating the influence of specific data while maintaining overall model Utility.

To ensure a fair comparison, we replicated the original model's training process as closely as possible. This included maintaining the same architecture (ResNet-18), hyperparameters, dataset preprocessing techniques, and optimization settings. By keeping all conditions identical (except for the removal of the Forget Set) we ensure that significant differences in performance can be attributed primarily to the impact of forgetting rather than variations in training methodology. Although minor variation may occur due to parameter initialization, experimental results indicate that the impact is minimal.

The reference model serves as a best-case reference, allowing us to assess how closely different unlearning techniques approximate the ideal outcome—namely, a model that behaves as if the target data had never been seen. While it is not possible to guarantee complete removal of the forgotten data, Machine Unlearning is generally considered successful when the resulting model is highly similar or indistinguishable from a model trained from scratch without the Forget Set. Measuring this similarity, however, remains an open challenge, as there is currently no universally accepted standard. Various metrics have been proposed to address this issue, each capturing different aspects of model behavior, as discussed in Section 2.5.

3.3. Unlearning Models

As mentioned in the previous section, resource limitations played a significant role in determining which unlearning models to implement. For this reason, we chose to develop two unlearning models based on the fine-tuning, described in Sections 2.3 and 2.4. Additionally, since the main objective of this work is to compare various different Forget Set scenarios, starting with simpler approaches rather than more complex methods allows for a more manageable training process and facilitates result interpretation.

3.3.1 Basic Fine-Tuning

Basic Fine-Tuning is one of the simplest approaches to unlearning, where the model continues training for a few epochs after the Forget Set has been removed from the training data. Instead of enforcing forgetting through specialized mechanisms, this method relies on the model's natural adaptation to the modified dataset. The process follows these steps:

1. **Loading the Original Model:** We use the Original Model, which has been trained using the full CIFAR-10 dataset, ensuring it reaches an adequate accuracy before proceeding with unlearning.

2. **Defining the Forget Set:** The subset of data that must be forgotten is identified and removed from the training set.

3. **Creating the Retain Set:** The remaining training data, after removing the Forget Set, constitutes the new training set (Retain Set).

4. **Retraining with the Retain Set:** The model is fine-tuned exclusively on the Retain Set using a lightweight retraining procedure designed to promote forgetting while preserving performance. Specifically, training continues for 10 additional epochs with a reduced learning rate (0.01) using SGD with momentum and weight decay. A CosineAnnealingLR scheduler is employed to gradually decay the learning rate over the fine-tuning period. These settings aim to ensure that the model adjusts to the new data distribution without diverging too far from its original parameters, thus reducing its dependence on the forgotten samples while avoiding overfitting to the Retain Set.

While this approach can reduce the model's reliance on removed samples, it does not ensure complete forgetting. Traces of the forgotten data may persist within the learned representations, limiting the effectiveness of fine-tuning as a dedicated forgetting mechanism. Nevertheless, due to its simplicity, low computational cost, and reasonable performance, this method is widely regarded as a strong baseline in the Machine Unlearning literature, providing a practical point of comparison for evaluating more complex strategies [16].

In terms of expected outcomes, the model should retain accuracy on the remaining dataset, as most of the original training data remains unchanged. However, its performance on the Forget Set is expected to degrade relative to the original model, indicating partial unlearning. Ideally, the fine-tuned model would approximate the behavior of a fully reference model on the Retain Set in terms of forgetting while preserving as much useful knowledge as possible.

3.3.2 Basic Fine-Tuning with Perturbation

This method follows the same procedure as Basic Fine-Tuning but introduces an additional step: resetting the parameters of the fully connected (FC) layer before retraining. By resetting the FC layer, we eliminate its previously learned weights, forcing the model to relearn the final mapping from features to class labels. This disruption is expected to initially degrade overall performance due to the reinitialization of the fully connected (FC) layer weights, but the model typically begins to recover accuracy after a few training steps as it readapts to the retained data. However, in some cases, it may aid unlearning by reducing reliance on pre-existing feature representations.

The reset is implemented by reinitializing the weights of the fully connected (FC) layer using Kaiming normalization for the weight parameters and setting the bias terms to zero. This modification allows us to study whether the forced adaptation of the final layer influences the model's forgetting behavior.

While our approach uses a straightforward perturbation focused on the FC layer, many other strategies could be considered. For example, one could reset weights in earlier layers, partially perturb subsets of neurons, or even introduce noise into specific feature maps. Exploring such perturbation schemes could yield more unlearning solutions, but this was beyond the scope of our study.

In this work, our goal was not to develop novel unlearning methods, but rather to focus on the behavior of different Forget Set strategies. As such, we chose to adopt relatively simple unlearning baselines, such as fine-tuning with and without perturbations, to serve as proof-of-concept mechanisms in order to highlight the potential and limitations of carefully designed Forget Sets. The impact of this reset is illustrated in Figure 4, which shows the parameter distribution before and after the reset.

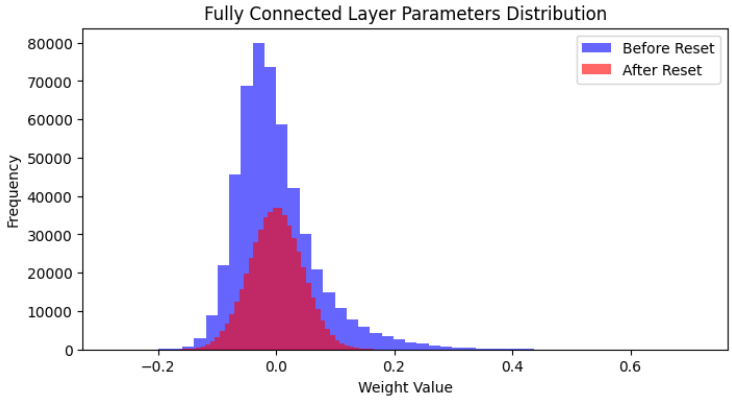

Figure 4: Distribution of the parameters in the final fully connected (FC) layer of the "Basic Fine-Tuning with Perturbation" model before and after the reset operation. This reset reinitializes the weights of the FC layer to break the direct memorization of class-specific features. As shown, the distribution becomes centered around zero after the reset, indicating that the parameters have been successfully reinitialized. This strategy is used in the Fine-Tuning with Perturbation method to enhance unlearning by encouraging the model to re-adapt its final decision boundaries.

3.4. Forget Sets Design

The primary objective of this study is to analyze and evaluate the effectiveness of different Machine Unlearning strategies, identifying the best, worst, and intermediate approaches that lead to strong, weak, or incomplete unlearning. A key aspect of this analysis is the precise definition of Forget Sets and the methodologies used to construct them.

To achieve this, we have defined and structured distinct Forget Sets, each representing a different scenario and following different criteria: Arbitrary Forget Set Selection (Section 3.4.1), Category-Based (Section 3.4.2), Confidence-Based (Section 3.4.3) and Similarity Density-Based Forget Set Selection (Section 3.4.4). Rather than directly comparing all sets at once, we have designed multiple experiments, each focusing on specific subsets of Forget Sets. These experiments allow us to systematically assess different strategies under varied conditions, providing a more comprehensive understanding of their effectiveness.

Each Forget Sets contain 5,000 samples selected from the original CIFAR-10 training set, ensuring that each unlearning task is non-trivial yet still feasible within the experimental scope. Fixing the Forget Set size across all experiments ensures fair comparison between strategies, as the challenge introduced by the removal remains consistent. At the same time, this size is large enough to potentially affect the learned model's performance significantly. This constraint thus highlights the impact of different selection criteria, allowing us to isolate the effects of Forget Set construction strategies on unlearning efficacy.

3.4.1 Arbitrary Forget Sets

Next, we describe the design of two Forget Sets by randomly selecting samples from the dataset without considering their significance, distribution, or relationship to the learned model. These selections are independent of any structured forgetting strategy, ensuring that the removed data points are chosen in an uninformed manner. The distribution of the Forget Sets is illustrated in in Figure 5.

Figure 5: Class distribution of Forget Sets 1 and 2, which are defined through random selection from the CIFAR-10 training data. These sets are constructed without considering sample importance, structure, or feature relevance, serving as a baseline for evaluating unlearning performance under uninformed data removal. The figure shows the number of samples per class in each set, highlighting the balance inherent to their construction.

As we can see in Figure 5, the distribution of the selected data is fairly similar and well-balanced. This is expected, as the random selection process ensures that all data points have an equal probability of being chosen, leading to a distribution that closely reflects the overall dataset. Since no bias or specific criteria were applied in the selection, the Forget Sets naturally inherit the statistical properties of the original data, resulting in comparable distributions. After defining the Forget Sets, we construct the corresponding Retain Sets by removing the data points belonging to the Forget Sets from the original training set. This ensures that the retained data does not contain any of the samples designated for forgetting.

With these Retain Sets, we proceed to train two new models from scratch (one for each Retain Set) that will be used as reference models to evaluate the unlearning process in a later stage, using the same architecture and general training setup as the original model trained on the full CIFAR-10 dataset. To ensure fairness in comparison, we apply consistent hyperparameters (e.g., optimizer settings, learning rate schedule, and number of maximum epochs). However, rather than training for a fixed number of epochs, we use early stopping based on validation loss to prevent overfitting and improve generalization.

In each case, the model with the lowest validation loss is saved and used for evaluation, rather than the model from the final epoch.

Furthermore, we apply standard data augmentation techniques during training (including random cropping with padding and horizontal flipping) to enrich the training data and improve robustness. This setup allows for a more reliable assessment of how well the model adapts to training without the Forget Set.

Once the models are trained, we apply two different Machine Unlearning techniques, Fine-tuning (FT) and Fine-tuning with Parameter Resetting (FTP), previously detailed in Sections 3.3.1 and 3.3.2, respectively. Note that the unlearning process is applied on the original model (M_o), and not on the two new models trained from scratch on the retained data, which will be used later in the evaluation process as reference models.

3.4.2 Category-Based Forget Sets

In this approach, two Forget Sets are constructed by removing all samples belonging to a single class from the CIFAR-10 training set. The selection of the categories is not random but based on the model's performance during training: one Forget Set contains all samples from the category with the highest training accuracy, while the other includes all samples from the category with the lowest training accuracy. This structured removal based on training accuracy provides a simple yet systematic way to define class-based Forget Sets.

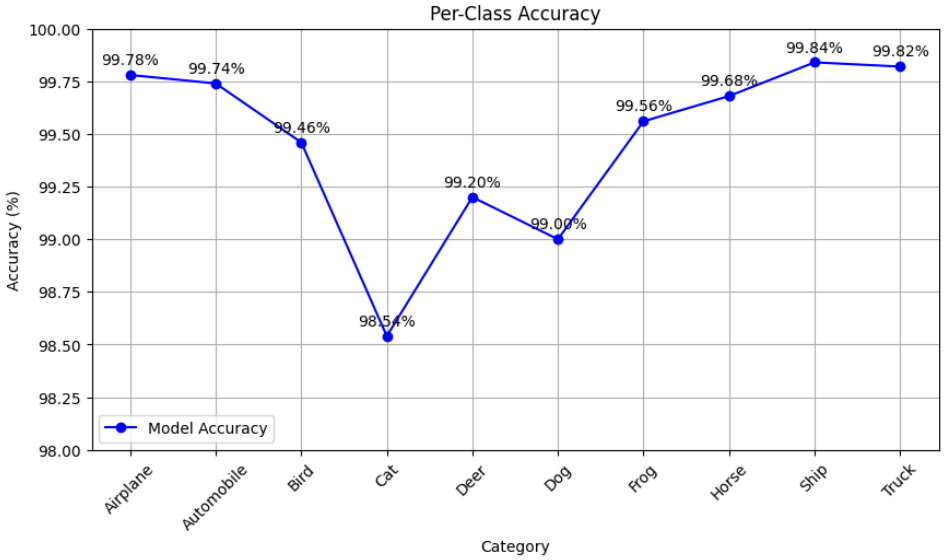

Figure 6: Per-class training accuracy of the original model (M_o).

Figure 6 shows the per-class training accuracy of the original model. As it can be observed, the category with the highest accuracy is "ship", while the lowest accuracy is observed in the "cat" class. However, the accuracy differences across all categories are minimal, suggesting that removing either the best or worst-performing category may not

drastically impact overall model performance.

3.4.3 Confidence-Based Forget Sets

The Forget Sets detailed in this section are constructed based on the model's confidence in its predictions over the training data. Specifically, we begin by evaluating all training samples using the pretrained ResNet-18 model. For each sample, we compute a confidence score defined as the softmax probability assigned to the predicted class—indicating how certain the model is about its classification. The softmax function transforms the raw output logits z_i of the model into a probability distribution over classes, and is defined as:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

where z_i is the logit corresponding to class i, and the denominator sums over all class logits z_j. This yields a probability between 0 and 1 for each class, with higher values indicating greater model confidence. The predicted class is the one with the highest softmax score, and its associated probability is used as the sample's confidence.

After computing these scores for all training samples, we sort the training set in descending order of confidence. From this ordered list, we then define two distinct Forget Sets. The *Best Confidence Forget Set* consists of the 5,000 samples with the highest confidence values—i.e., those the model is most certain about. Conversely, the *Worst Confidence Forget Set* includes the 5,000 samples with the lowest confidence scores, representing inputs that the model finds most uncertain.

By using confidence scores as the selection criterion, this method introduces a systematic and quantifiable way to explore how a model's certainty impacts the effectiveness of Machine Unlearning. Both sets are constrained to the same fixed size (5,000 samples) to maintain experimental consistency and comparability across different forgetting strategies.

3.4.4 Similarity Density-Based Forget Sets

To investigate the impact of interconnectivity between the forget and Retain Sets, we designed a methodology that quantifies this interconnection using a similarity-based approach. In this context, *interconnectivity* refers to the degree of structural or representational similarity between the data samples we wish to forget and the ones we intend to retain. A high interconnectivity implies that the Forget Set shares strong internal representations with the Retain Set, making the unlearning process potentially more difficult, as the model may generalize shared features. Conversely, a low interconnectivity would indicate that the Forget Set is structurally distinct, potentially making it easier to isolate and remove its influence from the model.

The recent work in [17] highlights this concept with the assertion that: *"Unlearning is harder when examples from the Forget Set are structurally entangled with the Retain Set, and easier when the Forget Set is structurally distinct."* This insight directly motivates our approach, in which we design Forget Sets based on varying levels of similarity to the Retain Set, aiming to empirically test how this structural entanglement impacts the forgetting performance.

To measure interconnectivity, we leverage the internal feature representations of the last Fully Connected layer of the pre-trained ResNet-18 model trained on CIFAR-10 (using the original model M_o, trained on the whole train set). By extracting these deep features for all training samples, we construct a similarity matrix that captures pairwise similarities between data points.

The similarity between samples is computed from the Euclidean distance between their feature representations, as detailed next. The Euclidean distance $d(\mathbf{x}_i, \mathbf{x}_j)$ between two feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$$

where x_{ik} and x_{jk} are the k-th components of the feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, and n is the number of dimensions in the feature space.

We then derive the similarity by inverting these distances, such that more similar samples have higher similarity scores. The similarity $\text{sim}(\mathbf{x}_i, \mathbf{x}_j)$ is calculated as the inverse of the distance:

$$\text{sim}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{1 + d(\mathbf{x}_i, \mathbf{x}_j)}$$

Then, the number of strong cross-class similarities is counted for each sample. If the count exceeds a predefined threshold T, the sample is marked as "eligible" for inclusion in the Forget Set. The threshold is defined as a percentage (i.e., 95%) of the maximum similarity observed across all training samples, ensuring that only samples with sufficiently strong connections to other categories are considered. Specifically, a sample i is considered eligible for inclusion in the Forget Set if the number of cross-class similarities S_i exceeds a predefined threshold N_1, defined as follows:

$$S_i = \sum_{j \in \mathcal{C}_i} \mathbb{1}(\text{sim}(\mathbf{x}_i, \mathbf{x}_j) > T)$$

where $\mathcal{C}_i$ is the set of samples from different classes (i.e., not belonging to the same category as sample i), T is the threshold based on the maximum similarity observed, and $\mathbb{1}(\cdot)$ is an indicator function that outputs 1 if the similarity between samples i and j exceeds the threshold, and 0 otherwise.

Additionally, a parameter N_1 (set to $N_1 = 33$ in our experiments) is used to specify the minimum number of cross-class similarities required for a sample to be considered eligible. This threshold was selected by computing the distribution of cross-class similarity counts across all training samples and adjusting N_1 such that the resulting eligible set comprises approximately 30% of the full training set. The eligibility of a sample i is defined as:

$$\text{Eligible}(S_i) = \begin{cases} \text{True}, & \text{if } S_i > N_1 \\ \text{False}, & \textit{otherwise} \end{cases}$$

Part of this process is illustrated in Figure 7, where a few samples are shown with the respective and most similar samples from other categories.

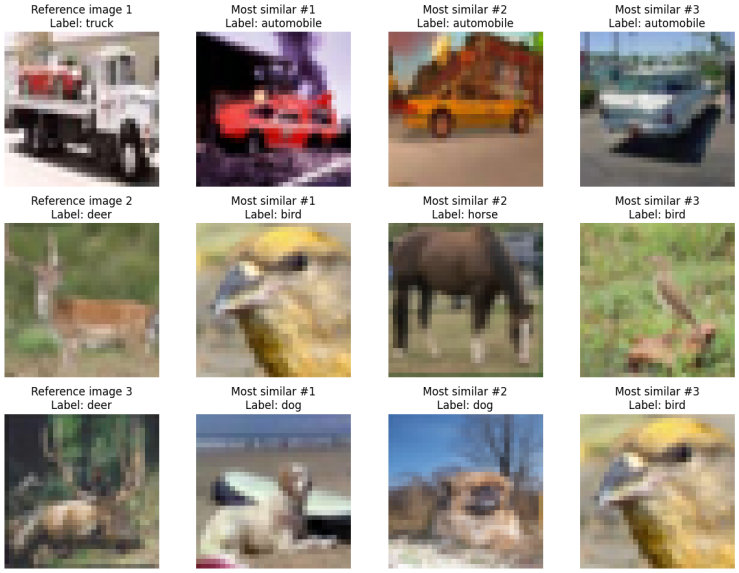

Figure 7: Examples of similar images across different categories. Each row starts with a reference image from the CIFAR-10 training set, followed by its three most visually similar images from other categories. As observed, these similar images belong to semantically related categories and share notable visual characteristics.

The process described so far allows us to isolate structurally entangled examples—those whose learned representations significantly overlap with multiple other classes. Moreover, we also consider the interconectivity between the Forget Set and the Retain Set, as mentioned before. Thus, we propose to create different Forget Sets, composed of images having strong cross-class similarities but also taking into account the amount of interconectivity with respect to the Retain Set, as detailed next:

- **Forget Set 7 (High Similarity Density)**: This set consists of 5,000 samples (100%) selected exclusively from the pool of eligible samples—those identified as highly interconnected based on their strong cross-class similarity. These examples exhibit strong similarity to other categories and are expected to be the hardest to forget, as there are many other similar in the Retain Set,
- **Forget Set 8 (Medium Similarity Density)**: This set contains 3,500 samples from the eligible group (70%) and 1,500 samples from the non-interconnected group (30%). It serves as a transitional case, balancing structural entanglement and separation, and allows us to assess how reduced feature-space entanglement between the forget and Retain Sets affects unlearning performance.
- **Forget Set 9 (Low Similarity Density)**: Composed of 2,000 samples from the eligible group (40%) and 3,000 non-eligible ones (60%), this set is dominated by structurally distinct examples. It is expected to represent the easiest unlearning scenario among the three, as a majority of the target samples are relatively well-separated from the Retain Set in the feature space, though some degree of entanglement still remains.

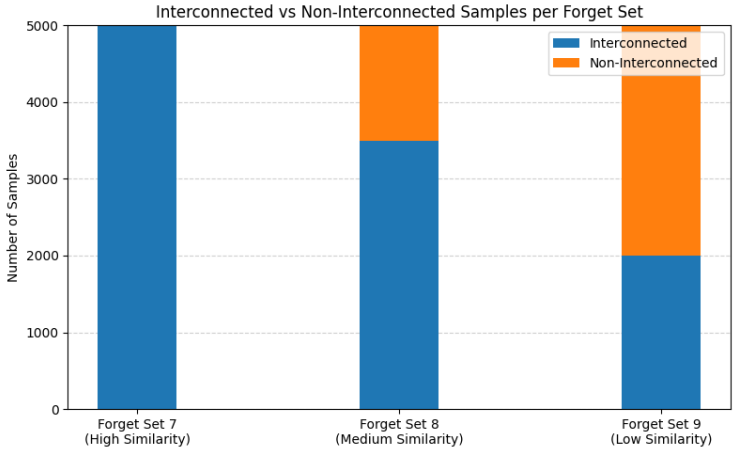

Figure 8: Distribution of interconnected and non-interconnected samples across the three Forget Sets. Each bar represents the composition of a Forget Set in terms of structural similarity density. Forget Set 7 consists entirely of highly interconnected samples, while Forget Set 9 contains a majority of non-interconnected ones, illustrating a gradient from theoretically complex to simpler unlearning scenarios.

Figure 8 illustrates the distribution of interconnected and non-interconnected samples across the three Forget Sets. As shown, Forget Set 7 contains only highly interconnected examples, indicating strong entanglement with the Retain Set. Forget Set 8 presents a mixed composition, with a majority of interconnected samples, while Forget Set 9 shifts the balance toward non-interconnected examples.

This three-tiered construction provides a controlled experimental setup for analyzing the role of interconnectivity between the Forget Set and the Retain Set in unlearning performance. By gradually decreasing the amount of interconnectivity of the Forget Set to the Retain Set, we can examine how the structural composition of the data affects the forgetting process.

3.5. Evaluation Metrics

Evaluation metrics are quantitative measures used to assess the performance of a machine learning model. In our analysis, we employed three metrics (Utility, Forgetting and Final Metric), which provide detailed insights into different aspects of the model's behavior. Notably, these metrics evaluate the model's ability to forget certain information (Forget Sets), its precision on the testing set and its retained Utility. These three metrics will be detailed next.

3.5.1 Accuracy

Accuracy is a commonly used evaluation metric in machine learning that measures a model's ability to correctly classify instances relative to the total number of instances

evaluated. It is defined as the ratio of correctly predicted samples to the total number of samples:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}}$$

Accuracy can be measured on the training, validation, and testing sets. Training and validation accuracy are typically monitored during the training process to track the model's progress and performance improvements. A higher accuracy on the testing set indicates that the model generalizes well, meaning it can correctly classify unseen data and is not simply memorizing the training samples. Good generalization is a fundamental goal in machine learning, as it ensures the model is robust and reliable in real-world applications. In our work, accuracy is used to compute Utility metric, as detailed next.

3.5.2 Utility

Let's define *Utility* as a way to evaluate how well a model retains its predictive performance after undergoing the unlearning process. Specifically, it measures the extent to which the model's accuracy aligns with that of a reference model M_r — the model trained from scratch using the Retain Set. By using this reference model as a benchmark, Utility metric provides insight into whether the unlearning method allows the model to continue making coherent and accurate predictions, despite the removal of certain data.

By evaluating Utility, we gain insight into the versatility of each model, allowing us to understand how well they can adapt to new or adjusted tasks while retaining their accuracy. This allows us to assess not only the effectiveness of the unlearning process but also the impact of various model characteristics on their overall robustness and adaptability.

Utility is computed by dividing the accuracy of an evaluated model (M_e) by the accuracy of a reference model (M_r), both evaluated on the test set D_t, as defined in Equation 3.1.

$$\text{Utility} = \frac{\text{Accuracy}(M_e, D_t)}{\text{Accuracy}(M_r, D_t)} \tag{3.1}$$

The value of *Utility* provides insight into the relative performance of the evaluated model compared to the reference model. If the Utility value is greater than 1 (Utility > 1), it indicates that the evaluated model outperforms the reference model on the test set, suggesting that it retained useful information and may generalize better. Conversely, if the Utility value is less than 1 (Utility < 1), the evaluated model performs worse than the reference model. A Utility value close to 1 (Utility ≈ 1) suggests that both models perform similarly, meaning that the evaluated model was able to maintain accuracy comparable to the reference model.

3.5.3 Forgetting

The *Forgetting* metric quantifies how much a model has "unlearned" specific data points after undergoing an unlearning process. This metric measures the difference between the outputs of the reference model and those of the evaluated model, allowing us to determine whether the model has effectively "forgotten" the designated Forget Set.

It is important to clarify that there is no single "correct" way to verify that a model has successfully forgotten, but we can assess whether its predictions are similar to those of the reference model. If the evaluated model produces outputs that closely resemble those of the reference model, this suggests that the unlearning process has been effective, as the model's behavior aligns with what would be expected if the data had never been learned in the first place.

The Forgetting value is computed using the Equation 3.2.

$$\text{Forgetting} = 1 - \frac{\sum_{i=1}^{|D_f|} d(M_r(x_i), M_e(x_i))}{|D_f| \times \sqrt{2}} \tag{3.2}$$

where:

- $|D_f|$: The total number of samples in the Forget Set D_f.
- x_i: A data sample from the Forget Set.
- $M_r(x_i)$: The output of the reference model when processing sample x_i.
- $M_e(x_i)$: The output of the evaluated model when processing sample x_i.
- $d(M_r(x_i), M_e(x_i))$: The Euclidean distance between the softmax output distributions of the reference model and the evaluated model for sample x_i. Since the softmax output represents a probability distribution (summing to 1), this distance measures how much the model's confidence in its predictions has shifted after unlearning.

To ensure the forgetting score lies within the range $[0, 1]$, we normalize the Euclidean distance by $\sqrt{2}$, which represents the maximum possible distance between two probability distributions in a 10-class softmax.

We apply $1 -$ normalized distance so that this value can later be interpreted as a weight: higher values indicate greater similarity to the reference model (i.e., more effective forgetting), while lower values reflect more substantial deviation (i.e., less effective forgetting). This formulation becomes particularly relevant in the next section, where we introduce the Final Metric that combines both forgetting and Utility components.

3.5.4 Final Metric (Combining Utility and Forgetting)

To obtain a comprehensive evaluation of a model's overall performance after undergoing an unlearning process, we define a Final Metric that combines the Utility and Forgetting metrics by multiplying them:

$$\text{Final Metric} = \text{Utility} \times \text{Forgetting}$$

This Final Metric allows us to assess both how much a model has forgotten and how well it retains its overall performance. A high value indicates that the model has successfully unlearned designated data while maintaining high accuracy, whereas a lower value suggests that the model either did not forget effectively or suffered a performance drop.

By integrating both aspects, this metric provides a holistic view of unlearning effectiveness.

Despite the utility of this aggregated measure, it is often necessary to analyze Utility and Forgetting separately to gain deeper insights. Since both components contribute equally to the final value, there can be cases where a model achieves a high Final Metric score by excelling in only one of the two aspects.

For example, a model could have significantly higher accuracy than the reference model (high Utility) but fail to properly unlearn certain information (low Forgetting). Conversely, another model might effectively forget the designated data (high Forgetting) but at the cost of a significant reduction in accuracy (low Utility). Depending on the specific application and priorities, one model might be more suitable than the other. In scenarios where unlearning is the top priority, a model with a slightly lower Final Metric but superior forgetting capabilities might be preferred. On the other hand, if maintaining predictive performance is critical, a model that retains high accuracy while forgetting less may be more desirable.

Thus, while the Final Metric serves as a useful global indicator, a detailed examination of both Utility and Forgetting independently is essential for making informed decisions based on the requirements of the specific task.

4. Experiments and Results

This chapter presents the empirical component of our work, structured to progressively analyze different strategies for constructing Forget Sets in the context of Machine Unlearning. The overarching goal is not to benchmark or develop new MU methods, but rather to identify and understand best/worst-case scenarios—specific Forget Sets that are particularly easy/hard for existing unlearning techniques to remove effectively. We begin by introducing the dataset and experimental setup. Then, we describe the experiments and results, which are designed to explore various Forget Set configurations. This will offer the necessary analysis to understand how different data characteristics impact unlearning performance and help shed light on the factors that define best/worst-case unlearning conditions.

For illustrative purposes, we also include the original model M_o in the result tables to support the analysis and discussion. However, it is important to note that, in the context of Machine Unlearning evaluation, there is no practical reason to directly compare any unlearning method with the original model since the original retains all training data and thus serves as a theoretical upper bound rather than a baseline for forgetting effectiveness.

4.1. Dataset: CIFAR-10

The CIFAR-10 [3] dataset (Canadian Institute for Advanced Research) is a widely used dataset in the fields of Computer Vision and Machine Learning. It consists of 60,000 RGB images, each with dimensions of 32x32 pixels. The images are distributed equally across 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck, with 6,000 images per class. Each image is labeled with its respective class, ensuring that it belongs exclusively to one category.

For our experiments, we utilized the predefined CIFAR-10 split, as described on the official dataset webpage (refer to the footnote for details), which is commonly used in the majority of experiments involving this dataset. This partitioning strategy allows us to prepare the data in a manner that aligns well with the goals of our project. By maintaining a clear and consistent separation between training, validation, and test sets, we minimize the risks of overfitting and data leakage. This not only ensures that our evaluation metrics provide a reliable and unbiased estimate of the model's generalization performance, but also facilitates reproducibility and fair comparison across different unlearning methods.

[3] CIFAR-10 dataset: `https://www.cs.toronto.edu/~kriz/cifar.html`

4.2. Experiment 1: Arbitrary Forget Set Selection

In this experiment, we explore the effects of defining Forget Sets using randomly selected data without any strategic criteria. The goal is to demonstrate that, although random selection of Forget Sets is a common practice in Machine Unlearning, it can cover meaningful differences in unlearning difficulty. This approach may accidentally mask best/worst-case scenarios, leading to overly optimistic or inconsistent conclusions about a method's effectiveness. Moreover, such variability complicates reproducibility and makes it harder to assess the robustness and general applicability of unlearning strategies.

The results obtained for the first two Forget Sets (randomly defined) are summarized in Table 4.1.

Table 4.1: Experimental results using two randomly selected Forget Sets (D_f). M_o represents the original model, trained on the whole train set; M_{ft} represents the model unlearned with a simple Fine-tuning (FT) strategy, whereas M_{ftp} using Fine-tuning (FTP) with perturbation, as detailed in Sections 3.3.1 and 3.3.2, respectively. All metrics use the reference model trained from scratch on the respective Retain set. The best and second best results are shown in bold and underlined, respectively, per case and per metric.

D_f	**Utility**			**Forgetting**			**Final Metric**		
	M_o	M_{ft}	M_{ftp}	M_o	M_{ft}	M_{ftp}	M_o	M_{ft}	M_{ftp}
Random (1)	**1.229**	<u>1.226</u>	1.215	<u>0.707</u>	**0.718**	**0.718**	0.870	**0.880**	<u>0.872</u>
Random (2)	**1.043**	<u>1.035</u>	1.031	0.852	**0.863**	<u>0.859</u>	0.852	**0.863**	<u>0.859</u>

The results presented in Table 4.1 reflect the impact of using randomly selected Forget Sets on the performance of unlearning strategies. Each model shows consistent behavior across both random sets, though the absolute metric values vary—highlighting the inherent unpredictability of random data removal.

From the perspective of Utility, the original model (M_o) performs best, as expected, since it retains access to the full training dataset, including the Forget Set. However, both fine-tuning strategies (M_{ft} and M_{ftp}) also achieve Utility values greater than 1 when compared to the reference model retrained without the Forget Set. This indicates that they retain strong performance on the retained data, despite the removal of the Forget Set. Among them, M_{ft} slightly outperforms M_{ftp} across both random Forget Sets, suggesting that the parameter perturbation introduced in FTP may lead to mild instability in the unlearning process.

Regarding the Forgetting metric, the original model consistently performs the worst, reaffirming its full retention of the forget data. Interestingly, both unlearning strategies achieve comparable forgetting scores in each Forget Set scenario, with M_{ftp} slightly edging out M_{ft} in one case. However, the differences are modest, suggesting that neither method is fully effective in eliminating traces of the forgotten data.

When looking at the final composite metric, which balances Forgetting and Utility, M_{ft} outperforms other models in both Forget Sets when analyzed individually, implying it provides the most favorable trade-off.

Overall, these results demonstrate that random selection of forget samples introduces considerable variance and complicates interpretability. This further motivates the need for more principled, adversarial, or influence-guided Forget Set design, which we explore in subsequent experiments.

4.3. Experiment 2: Category-Based Forget Set Selection

In this experiment, we explore the effects of defining Forget Sets by eliminating entire categories from the training set, as explained in 3.4.2. The goal is to analyze how removing semantically coherent groups of data impacts the unlearning process and whether the significance of the removed category (measured by its original training accuracy) affects the forgetting outcome.

Our initial hypothesis is that eliminating entire categories is not an effective way of evaluating unlearning strategies, as it may lead to severe disruptions in the model's internal feature representations. Machine Unlearning aims to remove specific information while maintaining the integrity of the remaining knowledge. However, deep learning models rely on shared features across categories, meaning that removing an entire class could distort the learned feature space, affecting generalization and stability.

The results obtained for these two Forget Sets are summarized in Table 4.2.

Table 4.2: Experimental results using two category-based Forget Sets (D_f). M_o represents the original model, trained on the whole train set; M_{ft} represents the model unlearned with a simple Fine-tuning (FT) strategy, whereas M_{ftp} using Fine-tuning (FTP) with perturbation, as detailed in Sections 3.3.1 and 3.3.2, respectively. All metrics use the reference model trained from scratch on the respective Retain set. The best and second best results are shown in bold and underlined, respectively, per case and per metric.

D_f	**Utility**			**Forgetting**			**Final Metric**		
	M_o	M_{ft}	M_{ftp}	M_o	M_{ft}	M_{ftp}	M_o	M_{ft}	M_{ftp}
Best Acc. (3)	**1.171**	<u>1.071</u>	1.036	0.099	<u>0.599</u>	**0.625**	0.115	<u>0.641</u>	**0.648**
Worst Acc. (4)	**1.117**	<u>1.039</u>	1.032	0.086	**0.620**	<u>0.616</u>	0.096	**0.645**	<u>0.635</u>

Starting with the Utility, all models maintain values above 1.0, indicating that their general performance on retained data remains good. The original model *Mo* achieves the highest Utility by design (**1.171** and **1.117**), as it retains the full training set, while both MU methods experience a slight improvement in Utility compared to the reference model. Among them, M_{ft} shows better Utility than M_{ftp} on both scenarios, reflecting its more conservative adaptation during fine-tuning. However, this comes with trade-offs in forgetting effectiveness.

The forgetting metrics reveal the core difficulty of this experiment. Forgetting entire categories proves far more challenging than removing randomly selected instances. The original model retains nearly all of its knowledge (obtaining very low forgetting scores, 0.099 and 0.086), while both MU strategies succeed in partially forgetting the removed categories. Yet, the maximum forgetting scores achieved (**0.625** by M_{ftp} when using Forget

Set 3 and **0.620** by M_{ft} when using Forget Set 4) remain moderate. This suggests that the model continues to retain significant latent information about the forgotten class, likely due to entangled representations in the feature space that are not easily erased without full retraining. Given that the fine-tuning process was conducted for a fixed number of epochs, one possible explanation is that the model did not have sufficient training iterations to fully adapt its internal representations and eliminate traces of the forgotten data

These results are further reflected in the Final Metric, yet their scores are limited by incomplete forgetting. The best result, **0.648**, is achieved by M_{ftp} in the Forget Set 3 case, indicating that perturbation helps dislodge tightly integrated class knowledge—but still not completely. The Forget Set 4, despite being a lower-performing category, is similarly difficult to unlearn, which makes sense because the difference in performance was very small.

Overall, these results emphasize a key limitation of current MU approaches: forgetting structured, semantically cohesive data such as full categories remains a difficult task. The high intra-class consistency of such groups creates deeply embedded patterns in the model, especially in its intermediate representations. Unlearning them may require more than minor updates—it demands substantial changes to the model's structure or distributional assumptions.

These findings ultimately validate our initial hypothesis: eliminating entire categories is not a reliable strategy for evaluating unlearning methods. Although useful as a stress test, it can introduce structural disruptions that distort the shared feature space across classes and compromises the goal of selectively removing information while preserving generalization. Future evaluations should include more fine-grained, realistic unlearning scenarios that better reflect the strengths and limitations of current MU techniques.

4.4. Experiment 3: Confidence-Based Forget Set Selection

In this experiment, we explore an alternative method for defining Forget Sets based on the model's confidence in its predictions (confidences being evaluated on the training set) as explained in Section 3.4.3. Unlike Experiment 2, where Forget Sets were constructed by eliminating entire semantic categories, this approach focuses on the certainty with which the model classifies individual samples. The goal is to assess how removing high-confidence versus low-confidence samples impacts the unlearning process and whether confidence-based selection provides a more effective strategy than category-based forgetting to support the evaluation of Machine Unlearning methods.

Our initial hypothesis is that forgetting high-confidence samples will result in greater overall forgetting compared to removing low-confidence samples. High-confidence samples correspond to well-learned and highly representative patterns that play a central role in the model's generalization. Removing such samples has the potential to disrupt fundamental feature representations, thereby inducing substantial forgetting across multiple related categories. However, it is important to consider that in the "Best Confidence" setting, the presence of similar samples remaining in the Retain Set may still support the model in maintaining those core representations.

In particular, forgetting high-confidence or highly representative examples (especially when they constitute a substantial portion of the data) could propagate broader changes throughout the model. However, the exact thresholds or conditions under which such disruption becomes critical remain unclear, suggesting an important direction for future work: to better understand how factors such as the number, confidence, or representational role of forgotten samples impact the effectiveness and stability of the unlearning process.

As in the previous experiments, the results are summarized in Table 4.3.

Table 4.3: Experimental results using two confidence-based Forget Sets (D_f). M_o represents the original model, trained on the whole train set; M_{ft} represents the model unlearned with a simple Fine-tuning (FT) strategy, whereas M_{ftp} using Fine-tuning (FTP) with perturbation, as detailed in Sections 3.3.1 and 3.3.2, respectively. All metrics use the reference model trained from scratch on the respective Retain set. The best and second best results are shown in bold and underlined, respectively, per case and per metric.

D_f	**Utility**			**Forgetting**			**Final Metric**		
	M_o	M_{ft}	M_{ftp}	M_o	M_{ft}	M_{ftp}	M_o	M_{ft}	M_{ftp}
Best Conf. (5)	**1.040**	<u>1.033</u>	1.031	<u>0.964</u>	<u>0.964</u>	**0.965**	**1.003**	<u>0.996</u>	0.994
Worst Conf. (6)	**1.047**	<u>1.043</u>	1.039	0.549	<u>0.597</u>	**0.601**	0.575	<u>0.622</u>	**0.629**

In this experiment, Utility scores remain slightly above 1.0 in both Forget Sets, indicating good performance on the retained data. As expected, the original model M_o, trained on the full training set, achieves the highest Utility scores (**1.040** and **1.047**). The MU models (M_{ft} and M_{ftp}) show similar results, with M_{ft} slightly outperforming M_{ftp} in both cases. This suggests that the fine-tuning-based updates introduce only minor improvements to overall performance, consistent with the goal of preserving retained knowledge during unlearning.

Turning to forgetting, the results appear, at first glance, to strongly support the initial hypothesis: removing high-confidence samples leads to significantly higher forgetting scores. For instance, in the "Best Confidence" setting, all models (including M_o) achieve remarkably high forgetting values (0.964–0.965), while in the "Worst Confidence" scenario, the forgetting scores are notably lower (0.549–0.601). However, these results require careful interpretation.

The unexpectedly high forgetting value given the adopted metric in the "Best Confidence" case do not necessarily indicate that the unlearning strategies were especially effective. Instead, this likely reflects the nature of the Retain set used to train the reference model. Because many high-confidence samples remained in the Retain set, as we predicted in our initial hypothesis, the retrained reference model closely resembled the original model M_o, which still contained the Forget Set. As a result, the measured forgetting (which compares outputs between the unlearned model and this reference) appears artificially high—not due to deep forgetting, but because both the unlearned and reference models are similar. This alignment creates the illusion of effective forgetting, even though core representations may remain largely intact.

In the "Worst Confidence" setting, forgetting scores are notably lower (0.549–0.601), indicating less effective forgetting according to the metric used. This likely stems from the fact that low-confidence samples tend to be less distinct and more sparsely represented, making it harder for the model to adjust its internal representations solely by removing these samples.

This also explains why the original model M_o achieves such surprisingly high forgetting scores in the "Best Confidence" condition. Since M_o still contains all samples (including those supposedly forgotten) it should, in principle, score poorly on forgetting. Yet, its high score (0.964) suggests that its predictions match the reference model's not because it forgot, but because the reference itself learned from many of the same high-confidence patterns still present in the Retain set.

Looking at the Final Metric, which balances Utility and forgetting, the "Best Confidence" setting again shows higher values (**1.003**, <u>0.996</u>, 0.994) than the "Worst Confidence" scenario (0.575, <u>0.622</u>, **0.629**). Here, too, the results suggest that the advantage observed in the high-confidence case is driven less by actual forgetting success and more by alignment with the reference model.

Overall, these results provide qualified support for our hypothesis that removing high-confidence samples induces greater forgetting score, given the adopted metric—but also reveal that the apparent forgetting may be overestimated due to shared information between the Retain set and the original model. These findings underscore the complexity of evaluating unlearning via reference-based metrics and highlight the importance of carefully selecting Forget Sets to avoid misleading conclusions.

This contrast becomes even more insightful when revisiting the previous experiment on category removal, where forgetting scores remained consistently low despite the scale of the intervention. That experiment showed how removing an entire class disrupted the model's structure without producing effective forgetting. In contrast, removing high-confidence examples, though smaller in scale, aligns better with how deep models encode information, targeting crucial points without destabilizing broader patterns.

In conclusion, while the metrics here initially suggest strong forgetting, a deeper analysis reveals that this is partly an artifact of overlap between training and reference sets. Still, the experiment reinforces a key takeaway: unlearning high-confidence, well-integrated knowledge is non-trivial, and its evaluation requires careful experimental design. Future protocols should ensure that Forget and Retain Sets are more distinctly separated to accurately reflect unlearning performance.

4.5. Experiment 4: Similarity Density-Based Forget Set Selection

This experiment further explores the impact of data entanglement on Machine Unlearning performance by evaluating Forget Sets constructed based on feature-space similarity. While the ideas presented in Section 2.6 broadly inspired our approach across all experiments, they are particularly relevant to the hypothesis tested here. By grouping and comparing data samples with high and low internal similarity, we aim to understand how structural properties of the data influence the effectiveness of unlearning methods.

In addition, it is strongly supported by recent developments such as the work presented in [17], a highly relevant and contemporary study that emphasizes the structural relationships within data in the context of unlearning.

The central idea behind this experiment is to evaluate the role of data interconnectivity in the effectiveness of the unlearning process. More specifically, the degree of similarity between the Forget Set and the Retain Set.

To this end, we construct three distinct Forget Sets characterized by their average similarity density (detailed in Sec. 3.4.4) with respect to the Retain Set: *High Similarity Density*, *Medium Similarity Density*, and *Low Similarity Density*. These similarity levels are computed based on the representation space of the model, using internal embeddings to capture how "close" or interconnected the Forget Set samples are to those we intend to retain.

The underlying hypothesis is that Forget Sets with higher interconnectivity with the Retain Set will lead to lower forgetting scores due to residual influence and feature overlap, making effective unlearning more challenging. Conversely, Forget Sets with lower interconnectivity with the Retain Set are expected to facilitate more effective unlearning, as they will be less entangled with the retained data. The goal of this experiment is to determine whether high similarity between the forget and Retain Sets impedes the unlearning process, potentially due to overlapping or entangled features, while low similarity might enable a more distinct and effective removal of the target data. By analyzing the forgetting metrics across these three configurations, we aim to uncover the extent to which interconnectivity between the forget and Retain Sets plays a decisive role in the model's ability to unlearn.

This analysis not only builds on prior theoretical findings but also strengthens the practical dimension of our study, offering insight into how the structural composition of data affects unlearning outcomes. The results of this experiment are expected to provide a deeper understanding of the internal dynamics of forgetting and help guide more principled strategies for Forget Set selection in future applications.

The results obtained are presented in Table 4.4, allowing us to assess how varying levels of interconnectivity influence the forgetting performance.

Table 4.4: Experimental results using two Similarity Density-Based Forget Sets (D_f). M_o represents the original model, trained on the whole train set; M_{ft} represents the model unlearned with a simple Fine-tuning (FT) strategy, whereas M_{ftp} using Fine-tuning (FTP) with perturbation, as detailed in Sections 3.3.1 and 3.3.2, respectively. All metrics use the reference model trained from scratch on the respective Retain set. The best and second best results are shown in bold and underlined, respectively, per case and per metric.

D_f	**Utility**			**Forgetting**			**Final Metric**		
	M_o	M_{ft}	M_{ftp}	M_o	M_{ft}	M_{ftp}	M_o	M_{ft}	M_{ftp}
High Sim. (7)	**1.179**	<u>1.172</u>	1.165	<u>0.662</u>	**0.678**	**0.678**	0.780	**0.794**	<u>0.790</u>
Medium Sim. (8)	**1.169**	<u>1.164</u>	1.157	0.728	**0.740**	<u>0.737</u>	0.851	**0.861**	<u>0.853</u>
Low Sim. (9)	**1.051**	<u>1.043</u>	1.038	0.843	<u>0.851</u>	**0.853**	<u>0.886</u>	**0.888**	<u>0.886</u>

Analyzing the results in Table 4.4, we observe a clear and consistent trend across all

three similarity-based Forget Sets. As the similarity density between the Forget Set and the rest of the Retain Set decreases (from high to medium to low) the forgetting performance improves accordingly. Specifically, the forgetting metric (M_{ft}) increases from 0.678 for the high similarity set to 0.740 and 0.851 for the medium and low similarity sets, respectively. This clearly and consistently confirms the hypothesis that interconnectivity between the forget and Retain Set plays a significant role in the effectiveness of Machine Unlearning: the more isolated or dissimilar the data we aim to forget is, the easier it is to unlearn it without negatively impacting the rest of the model's knowledge.

Interestingly, the Utility metrics (M_{ft}) follow a mild downward trend, indicating a slight degradation in retained performance as the similarity decreases. This is expected, as lower similarity may indicate that the data is more specialized or resembles outlier examples, which the model may deem less relevant for generalization. Nevertheless, the Final Metric consistently shows improvement.

These findings emphasize the importance of considering the structural relationships within the dataset when constructing Forget Sets. This underlines the importance of incorporating similarity-based strategies in real-world applications where data separation and interdependence significantly affect unlearning dynamics.

It is important to note that the current experiment should not be directly conflated with the previous one, as they explore different dimensions of the unlearning challenge. In the earlier experiment, we examined how sample-level confidence influences forgetting, with some indirect implications regarding similarity. However, the focus there was on how well-learned (i.e., high-confidence) samples affect the forgetting process—leading to high forgetting scores that were later interpreted with caution due to overlap between the original and reference models. In contrast, the present experiment explicitly controls and varies interconnectivity between Forget and Retain Sets based on feature-space similarity. Although both experiments touch upon the concept of overlap, they do so from fundamentally different angles, making them inherently non-comparable.

The results for the forgetting metric presented in Table 4.4 become even clearer when visualized graphically. In Figure 9, we isolate the forgetting values for the three similarity-based Forget Sets and observe how forgetting consistently improves as the similarity density decreases. This creates a smooth and ascending curve that visually reinforces the trend: the lower the interconnection between the forget and Retain Sets, the more effective the unlearning process becomes.

4.6. Discussion: Summary of Experiments

Across four systematically designed experiments, we explored different strategies for constructing Forget Sets in order to evaluate and challenge the capabilities of Machine Unlearning methods. Each experiment introduced a distinct perspective on Forget Set selection (random, semantic, confidence-based, and structural similarity-based), allowing us to dissect how different types of data affect forgetting dynamics.

- **Experiment 1**: This experiment assessed the baseline performance of unlearning methods using arbitrarily selected Forget Sets. Results across two randomly defined sets revealed substantial variability—while Utility remained high for all models,

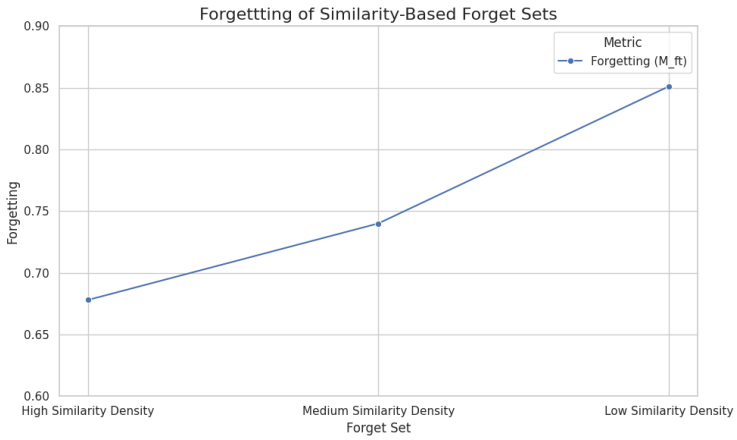

Figure 9: Forgetting performance across the three similarity-based Forget Sets, each constructed with a different level of interconnection to the Retain Set. The figure displays the evolution of the forgetting metric for each set, illustrating how lower similarity density leads to more effective forgetting.

forgetting effectiveness varied, with moderate scores not strongly influenced by the unlearning strategy. These findings underscore the limitations of arbitrary Forget Set construction, which lacks reliability and interpretability due to uneven influence of samples on the model's internal structure.

- **Experiment 2**: This experiment investigated the impact of forgetting entire semantic categories. While Utility remained reasonably high, forgetting scores were notably limited, likely due to deep feature overlap across classes. These results support the hypothesis that category-level removal, though intuitive, introduces structural disturbances that complicate evaluation and limit practical unlearning effectiveness.

- **Experiment 3**: This experiment examined Forget Sets based on model confidence, comparing high- vs. low-confidence sample removal. Eliminating high-confidence samples led to better forgetting scores, though this was partly due to alignment with the reference model. While Utility remained high, these results highlight the importance of careful forget-retain separation when using confidence as a selection criterion.

- **Experiment 4**: This experiment evaluated how increasing similarity between the Forget and Retain sets affects unlearning. Higher similarity consistently weakened forgetting, supporting the idea that structural entanglement makes unlearning harder.

These findings emphasize the need for improved unlearning benchmarks in classification tasks—benchmarks that go beyond simple class-based or random removals. They also highlight the importance of better defining and measuring overlap in learned representations, to advance both MU techniques and their evaluation metrics.

5. Conclusions and Future Work

This final section brings the project to a close by summarizing the main outcomes, reflecting on the challenges encountered, and outlining potential directions for future research. We begin with a set of general conclusions that assess how well the initial objectives were met and highlight the overall contributions of the work. We then review the specific findings related to each Forget Set strategy and hypotheses made. Following this, we discuss key limitations (both technical and methodological) that shaped the scope and execution of the study. Finally, we present proposals for future work, including improvements to the experimental setup, extensions to other domains, and avenues for developing more robust and scalable unlearning techniques. Together, these sections aim to consolidate the insights gained and provide a foundation for advancing research in Machine Unlearning.

5.1. General Conclusions

This project set out to investigate how the structure and selection of Forget Sets impact the effectiveness of different Machine Unlearning strategies. The primary goal was to understand which types of data are more difficult or easy to forget, and how to construct Forget Sets in a way that maximizes forgetting while minimizing disruption to retained knowledge. Additionally, this study aims to deepen our understanding of Machine Unlearning and contribute toward the design of more robust evaluation protocols, helping to assess MU methods more meaningfully.

Across all experiments, we systematically designed and analyzed a variety of Forget Sets, ranging from entire semantic classes to more fine-grained subsets selected based on model confidence or similarity between samples. This deliberate construction allowed us to compare unlearning performance under both extreme and subtle forgetting scenarios. Our findings highlight that the effectiveness of unlearning is highly dependent on the properties of the Forget Set itself and its relation to the retain set: not all data is equally forgettable.

These results underscore the critical importance of Forget Set design in any Machine Unlearning pipeline. A well-constructed Forget Set, aligned with the model's learned structure, can make the difference between a successful unlearning operation and one that disrupts the model without effectively removing the target information, with a direct impact on how MU models are evaluated.

In addition to evaluating forgetting effectiveness, we also developed and applied interpretable metrics to assess both Utility and Forgetting. These metrics enabled a rigorous, comparative analysis and revealed trade-offs inherent to different Forget Set strategies and unlearning techniques. The combination of simple yet informative evaluation tools and controlled Forget Set construction provided a solid foundation for our analysis.

In conclusion, all the original objectives of this work have been fulfilled. We demonstrated that Forget Set construction is a central factor in MU performance and evaluation, provided insights into which types of data are most resistant to forgetting, and laid out a reproducible methodology for analyzing unlearning behaviors. While the experiments were limited in scale due to computational constraints, the conclusions drawn from this study contribute to a more principled understanding of Machine Unlearning and establish a baseline for future research in this emerging field.

5.2. Key findings

The experiments collectively demonstrate that the difficulty of unlearning is strongly influenced by the structural properties of the Forget Set and its relation to the retain set. Arbitrary selection yields inconsistent results, as some samples are more entangled with the model's learned representations and harder to forget, highlighting the unreliability of random baselines. Forgetting entire semantic categories proves particularly challenging, as these groups often share deep feature representations with retained data, leading to limited forgetting despite significant interventions. Confidence-based selection shows that high-confidence samples are harder to unlearn in practice, though evaluation can be misleading when reference models share overlapping data. Most notably, similarity-based Forget Sets reveal a clear trend: the more structurally distinct the Forget Set is from the Retain Set, the more effective the unlearning process becomes. These findings validate that entanglement and interconnectivity between data points are key obstacles in Machine Unlearning, and that careful, structure-aware Forget Set design is essential for reliable evaluation and progress in the field.

5.3. Challenges and Limitations

Machine Unlearning remains an emergent and evolving field, characterized by a lack of standardized methodologies, scarce benchmarks, and limited consensus on evaluation practices. Its recent emergence has resulted in a fragmented landscape, with relatively few established strategies and minimal empirical validation across domains. This presents significant challenges for new research, as practitioners must navigate an underdefined space while designing, implementing, and evaluating their methods. Furthermore, the potential for MU to be applied across a wide variety of domains (such as computer vision, natural language processing, and multimodal learning) adds both to its appeal and its complexity, as it is unclear how generalizable current techniques are across different modalities and tasks.

Against this backdrop, one of the primary limitations of this project has been the availability of computational resources, which significantly influenced both the scope of the

unlearning techniques we could implement and the volume of experiments conducted. Our work was developed using Google Colab, which provides access to a free GPU with a memory cap of 15 GB. However, exceeding memory limits or prolonged execution time leads to abrupt session termination, requiring long wait times (often up to 12 hours) for the environment to become available again. These interruptions heavily delayed the training pipeline and constrained us to use lightweight, computationally efficient methods. Consequently, we had to limit the complexity of the models and the number of experimental configurations, despite the potential insights additional experiments might have provided.

Together, the novelty of the field and the technical limitations formed a compounded challenge. We often encountered implementation difficulties due to the scarcity of well-documented prior work, and resolving these issues required extensive trial-and-error and validation. This iterative process, although time-consuming, was essential to ensuring the methodological soundness of our approach within the constraints we faced.

5.4. Future Work

Future research and development in this project could follow multiple directions to enhance the effectiveness and robustness of the unlearning techniques explored. One of the primary areas for future work is the implementation of more advanced and computationally efficient unlearning methods, leveraging recent advancements in Machine Unlearning. Exploring techniques such as knowledge distillation [18], influence functions [19], or more sophisticated model pruning approaches [20] could significantly improve the balance between performance and forgetting efficiency.

Another crucial avenue is expanding the scope of the experiments by utilizing larger and more diverse datasets. Testing on high-dimensional, real-world datasets with greater variability in data distributions would provide more comprehensive insights into how different unlearning methods generalize across various domains. Moreover, going beyond image classification to include tasks such as natural language processing, multimodal learning, or structured prediction could expose different challenges and dynamics of Machine Unlearning, potentially requiring task-specific adaptations.

Furthermore, refining the definition of Forget Sets presents another important area for improvement. While our approach provides an initial framework for constructing and analyzing these sets, it represents only a small step toward understanding their full impact on unlearning performance. Since different strategies can be used to define what data should be forgotten, it would be valuable to explore multiple approaches. Investigating how the structure and selection of Forget Sets influence unlearning performance could help establish best practices for different use cases.

Beyond these experimental improvements, integrating unlearning verification techniques would also be an important step forward. Developing systematic methods to assess whether a model has genuinely forgotten data, potentially through explainability tools or adversarial validation, would enhance the reliability of unlearning methods.

By addressing these challenges, future work can contribute to making Machine Unlearning a more reliable, scalable, and practically applicable field, ensuring stronger pri-

vacy guarantees while maintaining high model performance.

Bibliography

[1] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480, 2015.

[2] B. Marino, M. Kurmanji, and N. D. Lane. Bridge the gaps between machine unlearning and ai regulation. *arXiv preprint arXiv:2502.12430*, 2025.

[3] H. Xu, T. Zhu, L. Zhang, W. Zhou, and P. S. Yu. Machine unlearning: A survey. *arXiv preprint arXiv:2306.03558*, 2023.

[4] W. Wang, Z. Tian, C. Zhang, and S. Yu. Machine unlearning: A comprehensive survey. *arXiv preprint arXiv:2405.07406*, 2024.

[5] C. Fan, J. Liu, A. Hero, and S. Liu. Challenging forgets: Unveiling the worst-case forget sets in machine unlearning. *arXiv preprint arXiv:2403.07362*, 2024.

[6] A. Golatkar, A. Achille, and S. Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. *arXiv preprint arXiv:2003.10933*, 2020.

[7] S. Neel, G. N. Rothblum, U. Stemmer, and S. Vadhan. Desiderata for unlearning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

[8] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[9] Y. Qu, X. Yuan, M. Ding, W. Ni, T. Rakotoarivelo, and D. Smith. Unlearning with neural networks: A comprehensive survey. *arXiv preprint arXiv:2305.07512*, 2023.

[10] David Zhang, Shidong Pan, Thong Hoang, Zhenchang Xing, Mark Staples, Xiwei Xu, Lina Yao, Qinghua Lu, and Liming Zhu. To be forgotten or to be fair: unveiling fairness implications of machine unlearning methods. *AI and Ethics*, 2024.

[11] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, and B. Zhang. Machine unlearning: A survey of algorithms and applications. *IEEE Access*, 2021.

[12] Z. Zuo, Z. Tang, K. Li, and A. Datta. Machine unlearning through fine-grained model parameters perturbation. *arXiv preprint arXiv:2401.04385*, 2024.

[13] Ajil Jalal Golatkar, Alessandro Achille, and Stefano Soatto. Initializing and fine-tuning neural networks with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[14] Chuncheng Guo, Tom Goldstein, Awni Hannun, Laurens van der Maaten Wu, Jonathan Wang, and Felix Yu. Certified data removal from machine learning models. In *International Conference on Machine Learning (ICML)*. PMLR, 2020.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] J. Ren, Z. Dai, X. Tang, H. Liu, J. Zeng, Z. Li, R. Goutam, S. Wang, Y. Xing, Q. He, and H. Liu. A general framework to enhance fine-tuning-based llm unlearning. *arXiv preprint arXiv:2502.17823*, 2025.

[17] H. Chang and H. Lee. Which retain set matters for llm unlearning? a case study on entity unlearning. *arXiv preprint arXiv:2502.11441*, 2025.

[18] Bai Li, Changyou Chen, and Lawrence Carin. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[19] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[20] Zhuang Liu, Mingjie Sun, Tianqi Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *International Conference on Learning Representations (ICLR)*, 2019.

[21] S. B. R. Chowdhury, K. Choromanski, A. Sehanobish, A. Dubey, and S. Chaturvedi. Towards scalable exact machine unlearning using parameter-efficient fine-tuning. *arXiv preprint arXiv:2301.03196*, 2023.

[22] M. Ding, J. Xu, and K. Ji. Why fine-tuning struggles with forgetting in machine unlearning? theoretical insights and a remedial approach. *arXiv preprint arXiv:2302.00356*, 2023.

[23] D. Zagardo. A more practical approach to machine unlearning. *arXiv preprint arXiv:2406.09391*, 2024.

[24] I. Premptis, M. Lymperaiou, G. Filandrianos, O. M. Mastromichalakis, A. Voulodimos, and G. Stamou. Ails-ntua at semeval-2025 task 4: Parameter-efficient unlearning for large language models using data chunking. *arXiv preprint arXiv:2503.02443*, 2025.

[25] Y. Jung, I. Cho, S.-H. Hsu, and J. Hockenmaier. Attack and reset for unlearning: Exploiting adversarial noise toward machine unlearning through parameter re-initialization. *arXiv preprint arXiv:2401.08998*, 2024.

[26] J. Yao, E. Chien, M. Du, X. Niu, T. Wang, Z. Cheng, and X. Yue. Machine unlearning of pre-trained large language models. *arXiv preprint arXiv:2402.15159*, 2024.

[27] A. Thudi, G. Deza, V. Chandrasekaran, and N. Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. *IEEE*, 2022.