$\nabla\tau$: Gradient-based and Task-Agnostic Machine Unlearning

Daniel Trippa [1] **Cesare Campagnano** [1] **Maria Sofia Bucarelli** [1] **Gabriele Tolomei** [1] **Fabrizio Silvestri** [1]

Abstract

Machine Unlearning, the process of selectively eliminating the influence of certain data examples used during a model's training, is a crucial area of research for safeguarding User Privacy and ensuring compliance with recent data protection regulations. Existing unlearning methods face critical drawbacks, including their prohibitively high cost, often associated with a large number of hyperparameters, and the limitation of forgetting only relatively small data portions. This often makes retraining the model from scratch a quicker and more effective solution.

In this study, we introduce **Gradient**-based and **T**ask-**A**gnostic Machine **U**nlearning ($\nabla\tau$), an optimization framework designed to remove the influence of a subset of training data efficiently. It applies adaptive gradient ascent to the data to be forgotten while using standard gradient descent for the remaining data. $\nabla\tau$ offers multiple benefits over existing approaches. It enables the unlearning of large sections of the training dataset (up to 30%). It is versatile, supporting various unlearning tasks (such as subset forgetting or class removal) and applicable across different domains (images, text, etc.). Importantly, $\nabla\tau$ requires no hyperparameter adjustments, making it a more appealing option than retraining the model from scratch. We evaluate our framework's effectiveness using a set of well-established Membership Inference Attack metrics, demonstrating up to 10% enhancements in performance compared to state-of-the-art methods without compromising the original model's accuracy.

[1]Sapienza University of Rome, Rome, Italy. Correspondence to: Daniel Trippa <daniel.trippa@gmail.com>, Maria Sofia Bucarelli <mariasofia.bucarelli@uniroma1.it>, Cesare Campagnano <cesare.campagnano@uniroma1.it>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

The field of machine learning has seen remarkable advancements in the past years. Current state-of-the-art models sometimes achieve performance levels comparable to human beings and obtain excellent accuracy in downstream tasks (LeCun et al., 2015; Silver et al., 2016; Brown et al., 2020; Gilardi et al., 2023). On the downside, deep machine learning models' growing complexity and scale introduce safety concerns ranging from User Privacy, potential biases in prediction, to deliberate output manipulation, among other issues (Shokri et al., 2017; Yeom et al., 2018; Li et al., 2020; Siddique et al., 2024). Indeed, publicly available models often rely on user-provided data and require adherence to the latest GDPR regulations, colloquially referred to as "right to be forgotten", allowing users to request the removal of their data from trained models for *privacy* reasons. Additionally, models trained on large amounts of publicly available data may encounter challenges in filtering and human-checking, potentially leading to *biases* if the data contains toxic or inappropriate content (Siddique et al., 2024). Consequently, there is a need to remove these biases once they are discovered. Meanwhile, threat actors have a rising presence who purposefully manipulate training data with malicious intent, causing arbitrary mispredictions when certain patterns are detected. This model behavior, known as a *backdoor* (Li et al., 2020), requires immediate removal of the influence of the manipulated data. All these scenarios, and many others, have in common the need to remove the influence that some training samples had on the final model.

A straightforward solution consists of starting a retraining procedure from scratch, excluding the data intended for removal. However, this is often impractical due to the significant time and computational power required.

Machine Unlearning addresses this challenge by developing methods to remove the influence of specific training samples efficiently, avoiding the need for complete retraining. Current works in the literature focus on different variations of this task, using distinct definitions based on their objective. Often, it is hard to compare all these approaches, as there is a lack of a precise definition and specific metrics. One method may be suitable for a certain instance of unlearning, such as "removing biases" (Yu et al., 2023), but may not work for other cases like "removing backdoors" (Liu et al., 2023).

In this context, we propose a comprehensive reinterpretation of the classic definition of approximating retraining, aiming for a more adaptable approach across various use cases. Our primary focus lies on enhancing User Privacy through unlearning. We emphasize the importance of defending against privacy-leaking attacks on specific training set samples.

Moreover, we aim to develop a method that is not overly sensitive to hyperparameters and does not require extensive experiments to find the best setup, as this time and computation resources could be otherwise used for retraining.

To address these objectives, we introduce **G**radient-based and **T**ask-**A**gnostic machine **U**nlearning ($\nabla\tau$), an optimization framework designed to remove the influence of training data subsets efficiently. $\nabla\tau$ demonstrates effectiveness across diverse unlearning scenarios while preserving the integrity of model performance. We conduct extensive evaluations of $\nabla\tau$ across different datasets and domains, including performing unlearning on models trained for image classification tasks on CIFAR-10 and CIFAR-100 (Krizhevsky, 2012), and text classification tasks on GoEmotion (Demszky et al., 2020). Finally, we investigate $\nabla\tau$'s performance across various hyperparameter values and various sizes of the set to be forgotten, demonstrating $\nabla\tau$ robustness. The main contributions of this work are the following:

- We present $\nabla\tau$, a method that adapts the gradient step to the quantity of information to be forgotten. The procedure introduced is both Model and Task-agnostic. Our method outperforms state-of-the-art methods and preserves accuracy levels present before unlearning.
- We conduct extensive experiments to prove the effectiveness of our method and compare it with other approaches over a heterogeneous set of setups, including different domains (Text, Image), unlearning tasks (Random subset removal, Class removal) and sizes of the forget set, up to 30% of the original training set.
- We perform a comprehensive evaluation of our method for different values of the sole hyperparameter introduced, providing insights on its correct use and empirically proving its robustness to small variations.

We publish our code at `https://github.com/dnl-trpp/Nabla-Tau`

2. Related Work

Several studies explore the concept of unlearning focusing on specific subtasks (e.g. removing bias, User Privacy) and currently a standardized definition is missing.

Unlearning for User Privacy. In this context, the goal is to mitigate the influence of specific sample subsets to safeguard data from privacy breaches like Membership Inference Attacks. Existing methods lack consistency in problem definition, framework, and evaluation. Graves et al. (2020) address unlearning as resistance to data-leakage attacks via label swapping. Chundawat et al. (2023) focus on removing forget set information using an incompetent teacher for approximation. Foster et al. (2023) propose a retraining-free method targeting influential weights, but it is limited to small forget set sizes. Kurmanji et al. (2023) introduce an unlearning method based on bad teaching effective in scenarios beyond User Privacy (e.g., removing biases), but evaluate their method only on class and subclass removal.

A drawback of existing methods is the introduction of new hyperparameters that require careful selection before unlearning, often making the process more time-consuming than simple retraining. Additionally, most studies focus exclusively on image classification models and small forget set sizes (less then 2% of training set).

Our method performs effectively in scenarios involving random subset removal, where forgetting samples do not necessarily share a common class or similarity. Additionally, we achieve optimal results for class removal. Our method proves effective for forget sets comprising up to 30% of the original dataset. We validate its applicability across image and text classification tasks. Importantly, our approach maintains remarkable robustness with respect to the sole introduced hyperparameter, crucial for practical use in real-world scenarios. We offer empirical insights on configuring this hyperparameter based solely on the unlearning set size.

Membership Inference Attack. Membership Inference Attacks (MIAs) (Yeom et al., 2018; Shokri et al., 2017) are privacy attacks on machine learning models where an attacker tries to determine whether a sample was used during training. In the context of Machine Unlearning, this attack is used as a metric to determine if the unlearning procedure can protect User Privacy for a given Forget Sample (Nguyen et al., 2022). In this work, we use the same MIAs used by Kurmanji et al. (2023) and Foster et al. (2023); in this setting, the attacker only observes model outputs. This is known as the Blackbox setting, contrasting with the Whitebox setting where the attacker can access all model parameters. Additional details will be included in Section 5.

We direct interested readers to the survey by Nguyen et al. (2022), which comprehensively covers various aspects and open questions of Machine Unlearning.

3. Problem Definition

Let D be a dataset and $\mathcal{A}$ a randomized training procedure. The output of $\mathcal{A}$, given D, and a fixed architecture is a vector of all model's parameters $\mathcal{A}(D) = w_o$. Due to the procedure's internal randomness, w_o is a random variable. We de-

note the function implemented by the model with parameters w as $f(\cdot, w)$, with a little abuse of notation, we will often refer to a model as its parameters. We define the *forget set* as a subset $D_f \subset D$ of samples from which we aim to *remove their influence* on the model. The *retain set* $D_r =$ is the complementary of the forget set. Given $w_o = \mathcal{A}(D)$, D_f and D_r, the goal of a deep Machine Unlearning procedure U is to produce a new set of weights $w_u = U(w_o, D_f)$ such that the 'unlearned model' w_u has 'forgotten' D_f without hurting the performance of the original model on $\mathcal{D}_r$. By 'forgetting' we mean the ability of the unlearned model to be indistinguishable under a certain metric from the golden baseline of retraining only on D_r. In *perfect unlearning* (Nguyen et al., 2022; Bourtoule et al., 2020; Brophy & Lowd, 2021; Thudi et al., 2022),we seek the distribution of models trained solely on $\mathcal{D}_r$ to match that of the unlearned model. Measuring these distributions is not trivial, and often, in specific applications, a less strict unlearning definition might suffice. Here, the focus is on preserving specific properties rather than the entire weights distribution. This is achieved by aligning certain metrics computed on the model's weights. More formally, for a map M that takes as input the model w and a subset of samples X in the input space $\mathbb{X}$ we require: $P(M(X, w_u) \in S) = P(M(X, w_r) \in S)\ \forall S \subset \mathbb{M}, \forall X \subset \mathbb{X}$.

represents the output space of the map M, and S is any measurable set in $\mathbb{M}$. If the map $M(X, w)$ is the output of the network with weights w on samples X, i.e., $M(X, w) = f(X, w)$, the problem is also referred to as *weak unlearning* (Baumhauer et al., 2022; Nguyen et al., 2022). The choice of M depends on the desired properties of the model to be preserved. For instance, in Bias removal, a metric measuring bias levels is used, while in User Privacy Unlearning, Membership Inference Attacks (MIA) metric are employed.

4. Our Method

We introduce a novel loss function aimed at eliminating the influence of samples in the forget set while maintaining the integrity of the model's performance. With a primary emphasis on User Privacy, our goal is to align the output distribution of samples within the forget set with that of the Test Set. We assume access to a validation set, but since we need only aggregate information of the set, such as its mean loss value even a small validation dataset, which is commonly available, is sufficient. If a validation set is not available, one could optimize the mean value of validation losses, potentially starting from the average training loss.Let L_D be the mean of the losses on a set D: $L_D = \frac{1}{|D|} \sum_{x \in D} l(f(x, w_i))$. We represent the mean loss on the forget set, retain set, and the validation set as L_{D_f}, L_{D_r}, and L_{D_v}, respectively. We introduce the following loss function:

$$L = \alpha(\text{ReLU}(L_{D_v} - L_{D_f}))^2 + (1 - \alpha)L_{D_r}.$$

The term $\text{ReLU}(L_{D_v} - L_{D_f})$ is used to reverse the gradient step on the forget set. This occurs only if the loss on the forget set is smaller than the objective loss L_{D_v}. Otherwise, the ReLU activation ensures that this term and its gradient become null and do not affect the optimization further. Optionally, the objective loss can be recomputed every c epochs. The parameter α balances the noise injection and the fine-tuning term, α equal to 0 corresponds to simple fine-tuning. Our findings suggest optimizing α using a scheduler yields the best results. Specifically, linearly decreasing α based on the number of optimization steps proves to be both efficient and fairly independent of the initial value of α. In Section Bof the Appendix, we experimentally demonstrate how to select an appropriate α based on the size of the forget set relative to the retain set. Deriving the loss we obtain:

$$\nabla L = \begin{cases} -\alpha 2(L_{D_v} - L_{D_f})\nabla L_{D_f} + (1-\alpha)\nabla L_{D_r} & L_{D_f} \leq L_{D_v} \\ (1-\alpha)L_{D_r} & L_{D_f} > L_{D_v} \end{cases}$$

The squared term $\text{ReLU}(L_{D_v} - L_{D_f})^2$ scales the negative gradient step ∇L_{D_f} in proportion to $L_{D_v} - L_{D_f}$, effectively creating an adaptive step.
An outline of our framework (Algorithm 1) and additional details are provided in Appendix A.

5. Experimental Setup

Our main focus is on forgetting a *random* subset of training data for User Privacy, where random means the samples have no correlation (e.g., these do not belong to the same class). This setting effectively represents scenarios where uncorrelated users request data removal. To validate our method, we conduct experiments on three classification datasets from different domains and perform additional tests to demonstrate robustness (Appendix B). With random subset removal being our primary focus, we also present results on Class Removal. All relevant material regarding this setting is provided in Appendix C.

Datasets and Architectures We evaluate our method across various datasets and domains. For image classification, we use the CIFAR-10 and CIFAR-100 benchmarks (Krizhevsky, 2012) following Chundawat et al. (Chundawat et al., 2023) with the ResNet18 (He et al., 2015) model. For text classification, we employ the GoEmotions dataset (Demszky et al., 2020), which labels sentences with 27 different emotions or as neutral, and use RoBERTa (Liu et al., 2019) with a linear layer on top as the model. Additional training details are provided in Appendix D.

Baselines We compare our method with state-of-the-art methods for random subset removal, particularly SSD (Foster et al., 2023) and Amnesiac (Graves et al., 2020). Additionally, we include the baseline of model Retraining, where the model is trained from scratch on the retain set, and simple Fine-tuning, where a pre-trained model is optimized

Table 1. Results on forgetting 3%, 15% of the CIFAR-10 train set and 15% of the CIFAR-100 train set. Mean and standard deviation values are averaged over three runs having different seeds. A_D is the model's accuracy on dataset D. MIA_L and MIA_E denote the MIA score using loss and entropy distribution, respectively. Bold font denotes the best results excluding the retraining golden baseline.

		$A_{D_r} \uparrow$	$\lvert A_{D_f} - A_{D_t} \rvert \downarrow$	$A_{D_t} \uparrow$	$\lvert MIA_L - 50 \rvert \downarrow$	$\lvert MIA_E - 50 \rvert \downarrow$
CIFAR-10 forget set 3%	Original	94.14± 0.00	9.24± 0.21	85.29± 0.21	5.13± 0.92	4.34± 1.02
	Fine-tuning	99.75± 0.01	5.93± 0.67	85.67± 0.52	2.47± 0.85	2.03± 0.71
	Retraining	95.30± 0.02	0.70± 0.52	84.07± 0.23	0.47± 0.31	1.70± 0.87
	SCRUB	94.11± 0.06	9.50± 0.10	85.19± 0.12	4.41± 0.60	3.50± 0.79
	SSD	94.14± 0.01	9.24± 0.20	85.31± 0.17	4.64± 0.64	3.44± 0.67
	Amnesiac	98.76± 0.09	18.87± 0.85	85.20± 0.16	11.20± 0.50	15.61± 0.51
	$\nabla\tau$ (*ours*)	**99.34± 0.07**	**2.06± 0.15**	**85.94± 0.46**	**0.60± 0.59**	**1.50± 0.70**
CIFAR-10 forget set 15%	Original	94.14± 0.00	9.13± 0.21	85.29± 0.21	4.71± 0.28	3.76± 0.17
	Fine-tuning	**98.77± 0.02**	6.30± 0.41	85.55± 0.30	3.02± 0.14	**1.57± 0.33**
	Retraining	86.98± 0.39	0.44± 0.17	80.25± 0.55	0.49± 0.25	0.74± 0.25
	SCRUB	93.90± 0.14	9.10± 0.13	85.02± 0.31	4.86± 0.20	3.67± 0.21
	SSD	94.13± 0.02	9.07± 0.17	85.34± 0.18	4.88± 0.20	3.76± 0.65
	Amnesiac	96.73± 0.06	4.22± 0.69	84.77± 0.24	4.73± 0.42	11.15± 0.49
	$\nabla\tau$ (*ours*)	97.82± 0.07	**2.39± 0.11**	**85.73± 0.25**	**1.52± 0.22**	**1.72± 0.28**
CIFAR-100 forget set 15%	Original	99.21± 0.00	39.12± 0.10	60.13± 0.10	24.24± 0.39	21.66± 0.32
	Fine-tuning	**99.97± 0.00**	31.83± 0.32	59.65± 0.25	16.08± 0.53	10.15± 0.14
	Retraining	77.84± 2.73	0.50± 0.33	50.60± 1.11	0.73± 0.32	0.28± 0.18
	SCRUB	96.05± 0.33	37.83± 0.28	58.13± 0.08	21.05± 0.69	17.47± 0.71
	SSD	99.14± 0.07	38.97± 0.02	**60.20± 0.03**	23.90± 0.26	21.44± 0.49
	Amnesiac	99.81± 0.03	**3.43± 1.28**	52.07± 0.33	5.37± 0.38	10.48± 0.54
	$\nabla\tau$ (*ours*)	99.74± 0.01	**3.38± 1.13**	58.39± 0.32	**0.95± 0.35**	**4.95± 0.50**

on the retain set. The starting model trained on the entire dataset is referred to as Original. Retraining is conducted for the same number of epochs as the Original model. For fair comparison, our method, Fine-tuning, and other baselines (excluding the training-free SSD) are run for 1/10 of the Retraining steps, ensuring an equal number of model updates.

Metrics To evaluate privacy leaks in the forget set of the final model, we use Membership Inference Attacks (MIA) as a metric (Kurmanji et al., 2023; Foster et al., 2023). The discrepancy between loss and entropy distributions of samples seen or unseen during training allows attackers (logistic classifiers) to infer sample membership using (see left side of Figure 1). We use the accuracy of an 'attacker' to evaluate Machine Unlearning. When the attacker is trained using the model's losses on forget and test examples, we denote its accuracy as MIA_L (Kurmanji et al., 2023); when using the model's output entropies on forget and test examples, we denote its accuracy as MIA_E (Foster et al., 2023). In the perfect scenario of Retraining the model only on D_r, the attacker's accuracy is 50%, equivalent to random guessing. We use accuracy as a metric to measure the efficiency and effectiveness of the final model. Our goal is to closely align forget set accuracy with test set accuracy to prevent attackers from inferring whether a sample from the forget set was used during training. We measure this alignment by calculating the difference between the unlearned model's accuracy on the forget set and its accuracy on the test set.

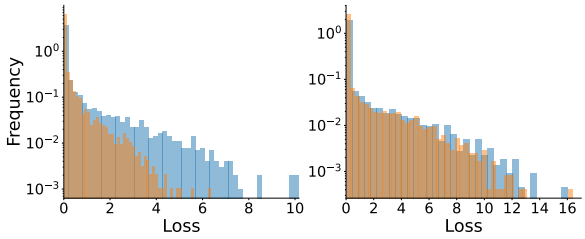

Figure 1. Loss distributions of Forget (orange) and Test (blue) sets before (left) and after (right) unlearning on CIFAR-10.

6. Experimental Results

We test our method across various scenarios to evaluate its effectiveness. We focus on unlearning for User Privacy, primarily in settings involving random subset removal, where the samples lack any specific correlation and we test our method with different forget set sizes, namely 3%, 15%, and 30% of the training set. Additionally, in Appendix C, we conduct experiments for the task of class removal. Finally, in Appendix B, we empirically demonstrate the robustness of our method with respect to its hyperparameter α.

Results of forgetting 3% and 15% of CIFAR-10 and 15% of CIFAR-100 are shown in Table 1. Complete results for

Table 2. Results on forgetting 3%, 15% and 30% of train set on GoEmotion for Text Classification. Results are obtained by averaging across three runs having different seeds. Bold font denotes the best results excluding the retraining golden baseline. A_D is the model's accuracy on dataset D. MIA_L and MIA_E denote the MIA score using loss and entropy distribution, respectively.

		$A_{D_r} \uparrow$	$\lvert A_{D_f} - A_{D_t} \rvert \downarrow$	$A_{D_t} \uparrow$	$\lvert MIA_L - 50 \rvert \downarrow$	$\lvert MIA_E - 50 \rvert \downarrow$
forget set 3%	Original	82.83± 0.03	29.00± 0.24	54.59± 0.08	15.13± 1.31	7.56± 1.73
	Fine-tuning	**61.17± 1.14**	**2.17± 1.25**	**53.87± 0.61**	**1.17± 0.93**	1.79± 0.94
	Retraining	85.82± 0.26	2.04± 0.92	54.45± 0.16	1.48± 0.83	1.24± 0.66
	$\nabla\tau$ (*ours*)	57.36± 1.83	2.84± 1.46	50.92± 1.15	1.29± 0.97	**1.19± 0.70**
forget set 15%	Original	82.85± 0.04	28.62± 0.38	54.61± 0.42	15.20± 0.65	7.19± 0.09
	Fine-tuning	**60.92± 0.43**	**3.72± 0.84**	**53.64± 0.64**	**1.18± 0.33**	1.08± 0.10
	Retraining	85.77± 0.12	0.63± 0.38	54.73± 0.12	0.39± 0.29	0.43± 0.18
	$\nabla\tau$ (*ours*)	**59.34± 0.83**	4.24± 2.18	52.17± 0.61	1.83± 0.97	**0.70± 0.17**
forget set 30%	Original	82.95± 0.02	28.36± 0.16	54.44± 0.20	15.01± 0.55	7.32± 0.16
	Fine-tuning	**62.85± 0.33**	**5.27± 0.23**	52.96± 0.22	**2.68± 0.06**	**0.33± 0.14**
	Retraining	84.80± 0.20	0.32± 0.34	54.17± 0.15	0.38± 0.35	0.44± 0.24
	$\nabla\tau$ (*ours*)	**62.59± 0.92**	6.32± 1.07	**53.22± 0.92**	2.93± 0.14	0.90± 0.48

CIFAR-10 are in Table 7 in Appendix, and for CIFAR-100 in Table 6 in the Appendix. Both datasets, CIFAR-10 and CIFAR-100, are utilized for image classification but exhibit significant differences. CIFAR-10 has fewer classes and a smaller initial difference between its test and forget sets, leading to a lower starting MIA compared to CIFAR-100. Results for the Text Classification task domain are presented in Table 2. Given the absence of evaluated state-of-the-art methods in this specific setting, we solely compare our method against retraining and fine-tuning baselines.

Results Our experiments revealed that for both Image Classification tasks, SSD (Foster et al., 2023) and Amnesiac (Graves et al., 2020) did not effectively reduce the MIA accuracy on forget sets of the tested sizes. SCRUB (Kurmanji et al., 2023), which focuses on class and subclass removal, demonstrates ineffectiveness in this setup. Amnesiac (Graves et al., 2020) significantly reduces forget set accuracy, but has a minor impact on MIA accuracy compared to the original model. In some cases, it performs worse than the original model (see CIFAR-10 experiment with a forget set size of 3%). SSD (Foster et al., 2023) has no effect on accuracy and shows no noticeable impact on MIA scores. As expected, fine-tuning achieves high accuracies on D_r and D_t, but offers no guarantee on forgetting and often performs poorly on MIAs. In contrast, our method surpasses other approaches in reducing the MIA score, nearly matching retraining baselines for both loss and entropy distributions. Additionally, our approach maintains consistently high Test Accuracy, sometimes even exceeding the initial accuracy, particularly in forget set sizes of 3% and 15% on CIFAR-10.

Also in text classification, our method effectively removes the influence of Forget Samples and defends against MIAs, even with a high initial difference between Train and Test set accuracy (27.3%). Notably, simple fine-tuning already provides effective defense against MIAs in this scenario.

7. Conclusions

In this work, we introduced $\nabla\tau$, a novel approach for conducting Machine Unlearning, specifically targeting the removal of a substantial subset of training data influence from the final model. Our method places a primary emphasis on User Privacy, this is especially relevant because of the "right to be forgotten" of the GDPR regulation, which requires data holders to be capable of removing user data upon request. We demonstrate its optimal performance measured in terms of accuracy and MIA, where the latter serves as a more reliable way of ensuring the model has forgotten the desired data, compared to accuracy alone. The effectiveness of our method has been validated across various settings, both in Image and Text classification tasks, outperforming several baselines. Notably, our approach showcases the capability to realign the distribution of the forget set closely with the Test Set, making our model excel in Machine Unlearning for User Privacy. Importantly, our approach introduces only one hyperparameter and demonstrates robustness without extensive tuning, crucial for real-world applicability. Although our experiments primarily focused on classification tasks, our method is task-agnostic, relying solely on loss alignment, and adaptable to diverse downstream tasks. In future works, we aim at exploring the adaptability of our method to additional tasks and domains. Moving ahead, our hope is that this research will establish a new foundation for investigating unlearning with a focus on User Privacy. Subsequent research should prioritize the standardization of Unlearning settings and definitions while exploring innovative methods adaptable to a variety of Unlearning scenarios.

Acknowledgements

This work was partially supported by the projects FAIR (PE0000013), SERICS (PE00000014), and SoBigData.it

(IR0000013) under the National Recovery and Resilience Plan funded by the European Union NextGenerationEU, as well as HypeKG – Hybrid Prediction and Explanation with Knowledge Graphs (H53D23003700006) under the PRIN 2022 program funded by the Italian MUR. Aslo supported by the project NEREO (Neural Reasoning over Open Data) project funded by the Italian Ministry of Education and Research (PRIN) Grant no. 2022AEFHAZ.

References

Baumhauer, T., Schöttle, P., and Zeppelzauer, M. Machine unlearning: Linear filtration for logit-based classifiers. *Machine Learning*, 111(9):3203–3226, 2022.

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., Lie, D., and Papernot, N. Machine unlearning, 2020.

Brophy, J. and Lowd, D. Machine unlearning for random forests. In *International Conference on Machine Learning*, pp. 1092–1104. PMLR, 2021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chundawat, V. S., Tarun, A. K., Mandal, M., and Kankanhalli, M. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher, 2023.

Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A. S., Nemade, G., and Ravi, S. Goemotions: A dataset of fine-grained emotions. *CoRR*, abs/2005.00547, 2020. URL https://arxiv.org/abs/2005.00547.

Foster, J., Schoepf, S., and Brintrup, A. Fast machine unlearning without retraining through selective synaptic dampening, 2023.

Gilardi, F., Alizadeh, M., and Kubli, M. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120 (30), July 2023. ISSN 1091-6490. doi: 10.1073/pnas.2305016120. URL http://dx.doi.org/10.1073/pnas.2305016120.

Graves, L., Nagisetty, V., and Ganesh, V. Amnesiac machine learning, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Krizhevsky, A. Convolutional deep belief networks on cifar-10. 05 2012.

Kurmanji, M., Triantafillou, P., Hayes, J., and Triantafillou, E. Towards unbounded machine unlearning, 2023.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Li, Y., Wu, B., Jiang, Y., Li, Z., and Xia, S. Backdoor learning: A survey. *CoRR*, abs/2007.08745, 2020. URL https://arxiv.org/abs/2007.08745.

Liu, J., Xue, M., Lou, J., Zhang, X., Xiong, L., and Qin, Z. Muter: Machine unlearning on adversarially trained models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4892–4902, October 2023.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach, 2019.

Nguyen, T. T., Huynh, T. T., Nguyen, P. L., Liew, A. W.-C., Yin, H., and Nguyen, Q. V. H. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models, 2017.

Siddique, S., Haque, M. A., George, R., Gupta, K. D., Gupta, D., and Faruk, M. J. H. Survey on machine learning biases and mitigation techniques. *Digital*, 4(1):1–68, 2024. ISSN 2673-6470. doi: 10.3390/digital4010001. URL https://www.mdpi.com/2673-6470/4/1/1.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Thudi, A., Deza, G., Chandrasekaran, V., and Papernot, N. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pp. 303–319. IEEE, 2022.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting, 2018.

Yu, C., Jeoung, S., Kasi, A., Yu, P., and Ji, H. Unlearning bias in language models by partitioning gradients. pp. 6032–6048, 01 2023. doi: 10.18653/v1/2023.findings-acl.375.

A. Additional Details on the Method

Algorithm 1 $\nabla\tau$ Training Loop

Input: Model, forget set, Validation Set, retain set,α
Output: Updated Model

for *each forget epoch* **do**
  if *n_epoch % c == 0* **then**
    L_{D_v} = `ComputeMeanLoss(`*Validation Set*`)`;
  end
  for *each* X_f,Y_f *in forget set* **do**
    X_r,Y_r = `NextRetainBatch()`;
    L_{D_f} = `Loss(`X_f,Y_f`)`;
    L_{D_r} = `Loss(`X_r,Y_r`)`;
    $L = \alpha\texttt{ReLU}((L_{D_v} - L_{D_f})^2) + (1-\alpha)L_{D_r}$
    `OptimizationStep(`*Model*,L`)`;
  end
  α = `SchedulerStep(`α`)`
end
return *Model*

Implementation insights: Balancing retain and forget sets An outline of our framework can be found in Algorithm 1. In practice, the forget set is typically smaller than the retain set. Each optimization step in our method accesses a batch from both sets, with the same batch size. Consequently, due to their size disparity, the Forget epoch — representing the number of steps required to process all forget set batches — differs from an epoch on the retain set.

B. Choosing α

Setup To test the robustness of our method, we repeat the procedure over many different forget set Sizes and with different values of our only hyperparameter α. We repeat each experiment with three different seeds and showcase the mean value. We plot the absolute difference from the perfect value 50% in a heatmap shown in Figure 4. A score close to 0 is a good defense against Membership Inference Attacks and approximates well the retraining baseline. Ideally, we want to observe a good score independently of the chosen hyperparameter α.

Results As observed in Figure 2, for each split of D_f there exist multiple values of α for which we obtain an optimal MIA score. This hyperparameter can be chosen in a range of values that all lead to good results. For instance, for a forget set size of 15%, all α values between 0.2 and 0.45 obtain values within the standard deviation of the golden baseline of retraining (0.5). Even when choosing an α that is outside this range of values, it can be observed that the results are less than 3% away from the values obtained by retraining.

We also visualize the absolute difference between the accuracy on D_f and the accuracy on D_t in Figure 3 . Consistent with MIA score observations, our method shows promise with sufficiently high α. Examining these findings reveals a positive correlation between discrepancies in accuracy and MIA scores.

As a rule of thumb, it can be derived by this experiment that a starting alpha that is around $\frac{5}{3}$ of the forget set size returns overall better performances across all the settings.

C. Class Removal

Setup When performing Class Removal together with the baselines of *Retraining* and *Fine-tuning* we compare our method with *SCRUB* (Kurmanji et al., 2023) that was specifically tested in this setting.

For class removal we conduct experiments focused on removing the influence of an entire class of CIFAR10. We test for two classes: “automobile” (Class 1) and “dog” (Class 5). We exclude the forgotten class from the Test set to evaluate the accuracy. The MIA is computed on the loss (and entropy) only on samples belonging to the class we intend to forget, including both samples seen and unseen during training.

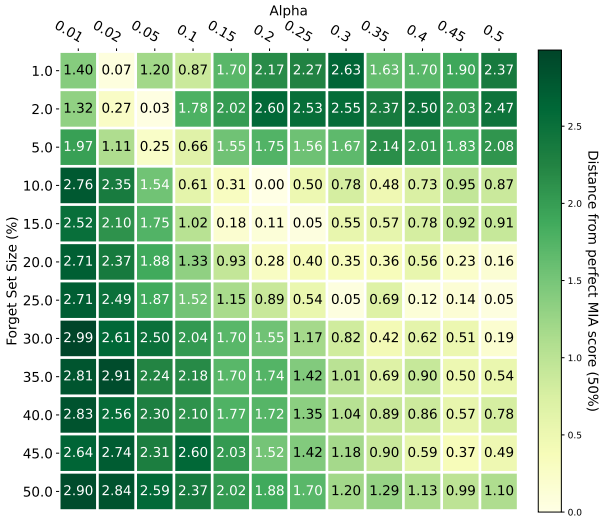

Figure 2. $|MIA_L - 50|$

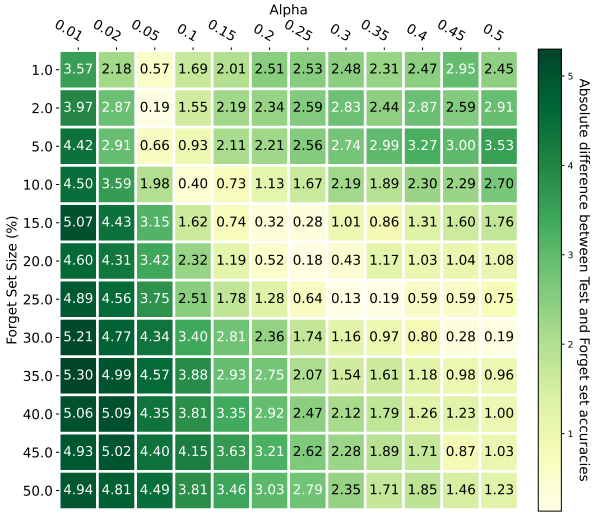

Figure 3. $|A_{D_f} - A_{D_t}|$

Figure 4. Experiment using our method on **CIFAR-10** across different forget set sizes (Y Axis) and α parameter (X Axis). On the left we report the absolute distance of MIA_L from the ideal value 50%. To improve the readability of the heatmaps, we do not report the standard deviations. However, for results where $|MIA_L - 50| < 1.0$ (the ones that best approximate retraining), the standard deviation is always under 1%. Note that even the golden baseline has a standard deviation $\pm 0.5\%$ from 50%. On the right, we report the absolute difference between the accuracy on forget set A_{D_f} and the accuracy on Test set A_{D_t}. The results highlight a similar pattern, indicating that similar scores in the accuracies are correlated to lower MIA scores. The results show the mean across three runs with different seeds.

Results The results presented in Table 3 demonstrate our approach's capability to entirely eliminate a specific class' impact. Accuracy on the forgotten class drops to 0, reproducing the results achieved through fine-tuning and the retraining baseline. The test accuracy remains consistently high, even surpassing the initial accuracy for class 5. Remarkably, the MIA score of our method notably decreases with respect to the original model, and $\nabla\tau$ outperforms SCRUB in almost all setups.

Table 3. Results on forgetting an entire class ("automobile" and "dog") of CIFAR10 for Text Classification. Mean and standard deviation are obtained by averaging across three runs with 3 seeds. Bold font denotes the best results, excluding the retraining golden baseline. A_D is the model's accuracy on dataset D. MIA_L and MIA_E denote the MIA score using loss and entropy distribution, respectively.

		$A_{D_r} \uparrow$	$A_{D_f} \downarrow$	$A_{D_t} \uparrow$	$\lvert MIA_L - 50\rvert \downarrow$	$\lvert MIA_E - 50\rvert \downarrow$
Class 1	Original	93.89± 0.00	96.46± 0.00	84.56± 0.22	2.75± 1.46	3.04± 0.98
	Fine-tuning	100.00± 0.00	0.00± 0.00	85.02± 0.20	1.68± 1.31	2.40± 1.38
	Retraining	100.00± 0.00	0.00± 0.00	76.46± 0.48	1.23± 1.01	1.39± 1.10
	SCRUB	**93.58± 0.10**	95.31± 0.14	**84.19± 0.22**	3.07± 0.30	**1.10± 0.89**
	$\nabla\tau$ (*ours*)	86.13± 18.50	**0.00± 0.00**	76.41± 13.18	**1.49± 1.49**	2.34± 0.67
Class 5	Original	94.90± 0.00	87.32± 0.00	86.49± 0.17	6.81± 0.47	2.55± 0.64
	Fine-tuning	100.00± 0.00	0.00± 0.00	88.21± 0.08	1.17± 0.62	1.06± 0.73
	Retraining	100.00± 0.00	0.00± 0.00	80.64± 0.30	1.99± 1.00	0.98± 0.47
	SCRUB	94.40± 0.04	89.46± 0.27	85.85± 0.10	7.73± 0.89	3.34± 1.95
	$\nabla\tau$ (*ours*)	**99.82± 0.01**	**0.00± 0.00**	**88.55± 0.13**	**2.14± 0.29**	**1.12± 0.80**

D. Additional Training Details and Hyperparameters

Table 4. Hyperparameters used during the training procedure on each Dataset. The resulting checkpoints are used for all following experiments on our unlearning procedure.

	CIFAR-10	**CIFAR-100**	**GoEmotions**
Architecture	ResNet18	ResNet18	RoBERTa
Optimizer	SGD	SGD	AdamW
Batch Size	256	256	128
Learning Rate	0.1	0.1	5e-5
LR decay	Linear [1,0.001]	Linear [1, 0.001]	Linear [1, 0.1]
Weight Decay	5e-4	5e-4	0.01

Table 5. Hyperparameters used during the unlearning procedure.

	CIFAR-10	**CIFAR-100**	**GoEmotions**	**Class Removal**
Optimizer	AdamW	AdamW	AdamW	AdamW
Batch Size	256	256	128	256
Learning Rate	0.001	0.001	0.0003	0.001
LR decay	Linear [1,0.1]	Linear [1,0.1]	N/A	Linear [1, 0.1]
Weight Decay	0.01	0.01	0.01	0.01
Alpha	5/3 * $\lvert D_f\rvert$	5/3 * $\lvert D_f\rvert$	0.5	5/3 * $\lvert D_f\rvert$
Alpha decay	Linear [1, 0]	Linear [1, 0]	Linear [1, 0]	Linear [1, 0]

To produce the starting checkpoint for the unlearning procedure on CIFAR-10 and CIFAR-100, we use data augmentations techniques including random crop and horizontal flip. Interestingly, we noticed that using data augmentation techniques during training helps reducing the MIA score of the resulting model. We decided to keep these checkpoints, as they provide a more realistic setting for a real-world unlearning use-case. All images are standardized by means and standard deviations.

In all our unlearning experiments, we use the AdamW optimizer with weight decay. During pre-training and subsequent experiments, we always pick the last model produced by the optimization procedure.

Table 4 provides a detailed description of the hyperparameters used in the pre-training procedures. The resulting checkpoints are used for all subsequent unlearning experiments.

Table 6. Results on forgetting 3%, 15%, and 30% of the CIFAR-10 train set. Mean and standard deviation values are averaged over three runs having different seeds. A_D is the model's accuracy on dataset D. MIA_L and MIA_E denote the MIA score using loss and entropy distribution, respectively. Bold font denotes the best results excluding the retraining golden baseline.

		$A_{D_r} \uparrow$	$\lvert A_{D_f} - A_{D_t} \rvert \downarrow$	$A_{D_t} \uparrow$	$\lvert MIA_L - 50 \rvert \downarrow$	$\lvert MIA_E - 50 \rvert \downarrow$
forget set 3%	Original	94.14± 0.00	9.24± 0.21	85.29± 0.21	5.13± 0.92	4.34± 1.02
	Fine-tuning	99.75± 0.01	5.93± 0.67	85.67± 0.52	2.47± 0.85	2.03± 0.71
	Retraining	95.30± 0.02	0.70± 0.52	84.07± 0.23	0.47± 0.31	1.70± 0.87
	SCRUB	94.11± 0.06	9.50± 0.10	85.19± 0.12	4.41± 0.60	3.50± 0.79
	SSD	94.14± 0.01	9.24± 0.20	85.31± 0.17	4.64± 0.64	3.44± 0.67
	Amnesiac	98.76± 0.09	18.87± 0.85	85.20± 0.16	11.20± 0.50	15.61± 0.51
	$\nabla\tau$ (*ours*)	**99.34± 0.07**	**2.06± 0.15**	**85.94± 0.46**	**0.60± 0.59**	**1.50± 0.70**
forget set 15%	Original	94.14± 0.00	9.13± 0.21	85.29± 0.21	4.71± 0.28	3.76± 0.17
	Fine-tuning	**98.77± 0.02**	6.30± 0.41	85.55± 0.30	3.02± 0.14	**1.57± 0.33**
	Retraining	86.98± 0.39	0.44± 0.17	80.25± 0.55	0.49± 0.25	0.74± 0.25
	SCRUB	93.90± 0.14	9.10± 0.13	85.02± 0.31	4.86± 0.20	3.67± 0.21
	SSD	94.13± 0.02	9.07± 0.17	85.34± 0.18	4.88± 0.20	3.76± 0.65
	Amnesiac	96.73± 0.06	4.22± 0.69	84.77± 0.24	4.73± 0.42	11.15± 0.49
	$\nabla\tau$ (*ours*)	97.82± 0.07	**2.39± 0.11**	**85.73± 0.25**	**1.52± 0.22**	**1.72± 0.28**
forget set 30%	Original	94.14± 0.00	8.93± 0.21	85.29± 0.21	5.05± 0.24	3.47± 0.38
	Fine-tuning	97.73± 0.05	6.72± 0.17	85.75± 0.33	2.90± 0.28	1.30± 0.18
	Retraining	95.17± 0.09	0.88± 0.40	82.24± 0.20	0.33± 0.39	0.46± 0.25
	SCRUB	94.09± 0.02	8.98± 0.24	85.17± 0.24	4.64± 0.12	3.63± 0.53
	SSD	94.14± 0.00	8.93± 0.21	85.29± 0.21	5.03± 0.13	3.62± 0.47
	Amnesiac	95.16± 0.07	**1.39± 0.09**	84.55± 0.11	**1.55± 0.92**	6.91± 0.23
	$\nabla\tau$ (*ours*)	**95.53± 0.12**	3.31± 0.14	**84.69± 0.05**	**1.70± 0.43**	**1.35± 0.04**

In Table 5 we present the hyperparameters used to conduct the experiments on our unlearning procedure. Notice that the *Retraining* baseline uses the same hyperparameters as pre-training, while *Fine-tuning* employs the same hyperparameters as shown in Table 5 (except for alpha hyperparameter that is not present in the regular Cross-entropy loss).

All the competing state-of-the-art baselines use the same hyperparameters provided by their official implementation. Our method and all the baselines, with exception of SSD, are based on a regular optimization procedure that runs unlearning for some epochs. SSD, instead, is a two steps *retraining-free* approach. For a fair comparison, all the other baselines – and our method – run for the same number of steps, equivalent to 6 epochs on the retain set.

E. Additional Experiments

Here, we provide the complete tables with the experiments conducted across various forget set sizes: 3%, 15%, and 30% for both CIFAR-10 and CIFAR-100. For the sake of completeness, we also include previously reported results, now divided into separate tables for CIFAR-10 Table 6 and CIFAR-100 Table 7.

Table 7. Results for forgetting 3%, 15%, and 30% of the CIFAR-100 train set across all baselines. Mean and standard deviation are obtained by averaging on three runs having different seeds. Bold font denotes the best results excluding the retraining golden baseline. A_D is the model's accuracy on dataset D. MIA_L and MIA_E denote the MIA score using loss and entropy distribution, respectively.

		$A_{D_r} \uparrow$	$\lvert A_{D_f} - A_{D_t} \rvert \downarrow$	$A_{D_t} \uparrow$	$\lvert MIA_L - 50 \rvert \downarrow$	$\lvert MIA_E - 50 \rvert \downarrow$
forget set 3%	Original	99.21± 0.00	39.13± 0.10	60.13± 0.10	22.63± 0.07	20.03± 0.20
	Fine-tuning	**99.96± 0.00**	31.22± 0.70	**60.09± 0.25**	15.22± 0.86	9.23± 0.77
	Retraining	98.77± 0.06	2.84± 0.68	53.49± 0.21	0.63± 0.53	0.13± 0.12
	SCRUB	96.65± 0.04	38.12± 0.36	58.33± 0.32	19.61± 0.51	16.42± 0.11
	SSD	98.82± 0.08	38.93± 0.16	59.94± 0.23	22.44± 0.22	19.97± 0.22
	Amnesiac	99.89± 0.01	34.28± 1.45	57.95± 0.28	11.98± 0.15	12.84± 0.08
	$\nabla\tau$ (*ours*)	99.91± 0.00	**3.78± 0.66**	**59.35± 0.15**	**0.88± 0.52**	**7.04± 0.59**
forget set 15%	Original	99.21± 0.00	39.12± 0.10	60.13± 0.10	24.24± 0.39	21.66± 0.32
	Fine-tuning	**99.97± 0.00**	31.83± 0.32	59.65± 0.25	16.08± 0.53	10.15± 0.14
	Retraining	77.84± 2.73	0.50± 0.33	50.60± 1.11	0.73± 0.32	0.28± 0.18
	SCRUB	96.05± 0.33	37.83± 0.28	58.13± 0.08	21.05± 0.69	17.47± 0.71
	SSD	99.14± 0.07	38.97± 0.02	**60.20± 0.03**	23.90± 0.26	21.44± 0.49
	Amnesiac	99.81± 0.03	**3.43± 1.28**	52.07± 0.33	5.37± 0.38	10.48± 0.54
	$\nabla\tau$ (*ours*)	99.74± 0.01	**3.38± 1.13**	58.39± 0.32	**0.95± 0.35**	**4.95± 0.50**
forget set 30%	Original	99.24± 0.00	39.02± 0.10	60.13± 0.10	24.07± 0.37	21.39± 0.37
	Fine-tuning	**99.97± 0.00**	32.11± 0.42	**59.63± 0.28**	16.32± 0.23	10.54± 0.23
	Retraining	99.22± 0.01	0.79± 0.48	49.22± 0.16	0.58± 0.13	0.63± 0.38
	SCRUB	96.85± 0.17	38.07± 0.14	58.48± 0.27	21.36± 0.49	17.97± 0.45
	SSD	99.24± 0.00	39.02± 0.10	60.13± 0.10	24.28± 0.44	21.34± 0.34
	Amnesiac	99.50± 0.07	17.32± 0.81	47.51± 0.35	11.62± 0.73	4.03± 0.36
	$\nabla\tau$ (*ours*)	98.02± 0.09	**5.89± 0.16**	54.48± 0.17	**2.74± 0.37**	**3.46± 0.18**