How to Make Reproducible Research in Machine Unlearning with ERASURE

How to Make Reproducible Research in Machine Unlearning with ERASURE

Andrea D'Angelo[1] , **Claudio Savelli**[2] , **Gabriele Tagliente**[1] ,
Flavio Giobergia[2] , **Elena Baralis**[2] , and **Giovanni Stilo**[1]
[1]University of L'Aquila,
[2]Polytechnic University of Turin
{andrea.dangelo6@graduate., gabriele.tagliente@student., giovanni.stilo@}univaq.it,
{claudio.savelli, flavio.giobergia, elena.baralis}@polito.it

Abstract

Machine unlearning, the process of removing specific data influences from Machine Learning models, is critical for complying with regulations like the GDPR's right to be forgotten and addressing copyright disputes in large models. Despite its rising importance, the field still lacks standardized tools, hindering reproducibility and evaluation. Here, we present, in an extensive way, ERASURE, a unified framework enabling reproducibility by implementing common unlearning techniques, evaluation metrics, and dedicated datasets. ERASURE advances research, ensures solution comparability, and facilitates reproducibility, addressing future legal and ethical challenges in data management.

1 Introduction

Rapid adoption of machine learning (ML) across industries has brought significant advances in automation, decision making, and data-driven insights. However, it has also introduced challenges related to data handling in compliance with legal and ethical standards. In particular, the *right to erasure* (or *right to be forgotten*) in regulations such as the General Data Protection Regulation (GDPR) requires that individuals' data be selectively removed upon request [Mantelero, 2013]. Upon such requests, the model's owner should build a new version without the removed data. However, retraining these models after every request is impractical due to the significant time, economic and environmental costs involved [Crawford, 2022], especially for large models. To overcome this problem, *machine unlearning* – the process of efficiently removing the influence of specific data points from a model – is emerging as a cornerstone of ethical AI, promoting compliance, accountability, and privacy in data-driven applications.

Machine unlearning is a growing field that currently lacks a widely accepted reference framework for collecting methods, metrics, and datasets. Some studies in the literature have attempted to address this limitation. For instance, in [Choi and Na, 2023], the authors provide the implementation of a limited number of existing unlearning techniques and metrics. However, the repository has not been designed to be an openly available library: the codebase does not allow easy extensions or usage in scenarios other than the replication of the experimental results of the paper. A further attempt at aggregating resources within machine unlearning is presented in [Nguyen *et al.*, 2022], where references to code repositories of the primary unlearning techniques presented in the literature are collected. However, this collection remains fragmented, highlighting the pressing need for a unified framework that provides a standardized interface for seamlessly adopting diverse unlearning techniques and scenarios.

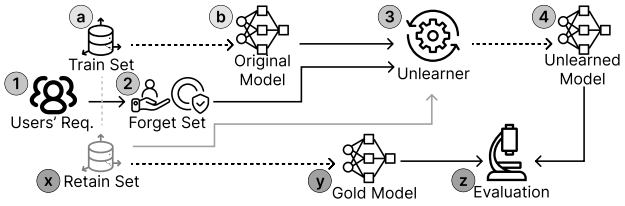

Figure 1: Extended Machine Unlearning Process: The dotted arrows depict training/unlearning operations. The main flow comprises steps from ① to ④. Steps ⓧ to ⓩ depict the evaluation.

To bridge this gap, we introduce *ERASURE* [1], a standardized evaluation framework for the machine unlearning process [Hayes *et al.*, 2024]. It provides ready-to-use implementations of various unlearning techniques, including retraining-based and approximate methods, along with metrics to assess *efficacy*, *utility*, and *efficiency* [Hayes *et al.*, 2024; Koudounas *et al.*, 2025]. ERASURE also offers curated datasets for benchmarking, ensuring adaptability across various use cases. By integrating existing solutions, datasets, and evaluation measures within a unified framework, ERASURE enhances reproducibility, accelerates advancements, and promotes alignment with regulatory standards.

To showcase ERASURE's flexibility and robustness, establishing it as a reference for future advancements in machine unlearning, this paper first introduces the Machine Unlearning Process, then it presents an overview of ERASURE. Section 3 presents use case scenarios through a main configuration, enabling reproducible experimental evaluations of various unlearning techniques.

[1]code available at `https://github.com/aiim-research/ERASURE`

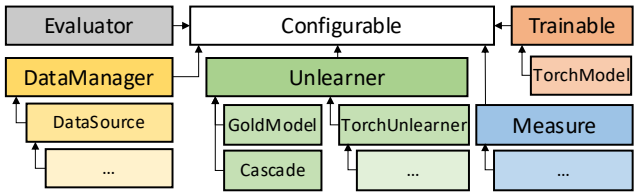

Figure 2: Overview of the main classes of the ERASURE Unlearning Framework and their relations.

2 ERASURE Unlearning Framework

Unlearning Process - The *Unlearning Process (UP)* typically adopted at production time comprises steps ① to ④ and uses the already available ⓐ and ⓑ as shown in Fig. 1. The UP has access to the *Original Model* ⓑ and its *Train Set* $\mathcal{D}$ ⓐ (e.g., an image classification model, trained on images published by users on a social network). The UP is triggered when a User's Unlearning Request ① is received. The Unlearning Request is identified by the *Forget Set* $\mathcal{D}_f$ ② – the specific set of instances that must be unlearned (e.g., all pictures published by a specific user). The *Retain Set*, $\mathcal{D}_r \triangleq \mathcal{D} \setminus \mathcal{D}_f$, ⓧ is the subset (grey dotted line on the left of Fig. 1) of the Train Set $\mathcal{D}$ that must not be deleted. Then, a specific *Unlearner* (i.e., Unlearning Method) ③ is applied to the Original Model to create the *Unlearned Model* ④, which must not have the influence of the Forget set instances. Note that some Unlearners can use only the Forget Set $\mathcal{D}_f$ while others also use the retain one $\mathcal{D}_r$ as depicted by the grey line in the middle of Fig. 1. To have an extended picture of the Machine Unlearning Process, refer to steps ⓧ to ⓩ as well. In this case, the primary interest is to precisely quantify the *quality* of the unlearning scenario that researchers and practitioners face (i.e., a company wants to test different unlearners on their datasets and solutions, or a researcher wants to test their method against the SotA ones). To do so, many measures in the SotA rely on the *Gold Model* ⓨ, which is the Original Model trained from scratch on the Retain Set. In the following, we will detail ERASURE design principles and its core components that help researchers and practitioners to meet their goals.

Design Principles - ERASURE is designed to support the needs of researchers and practitioners who aim to test and compare various unlearning techniques across multiple datasets, tasks, and models in a flexible and reproducible way. To meet full extensibility and flexibility, the framework follows object-oriented programming by grounding on abstract classes and fully embracing the factory pattern and the inversion of control paradigms. The companion of these principles is fully configurable workflows - shown in Sec. 3 - that provide robustness and precise control at the same time. The modular structure promotes easy integration of new datasets, unlearning methods, original models, and evaluation metrics. Moreover, to enhance reproducibility, ERASURE includes the implementation of the SotA unlearners, datasets, models, and measures, which can be used out-of-the-box for custom experiments by defining them in the main configuration file.

2.1 Core Components

ERASURE includes components (see Fig. 2) designed to develop and evaluate all the aspects of the unlearning process. Hereafter, we present the three main modules: data management, unlearning methods, and evaluation with measures.

Data Management - The *DatasetManager* orchestrates data handling by creating the *DataSource*, which is responsible for loading data from a specific source (e.g., files or repository). ERASURE provides built-in support for all the datasets available through the widely-used libraries HuggingFace [HuggingFace, 2025], TorchVision [TorchVision, 2025], and UCI Repository [Kelly *et al.*, 2025]. Once loaded, the data undergoes a series of – built-in or custom – preprocessing steps that are applied on the fly as batches are loaded. Developers can seamlessly integrate their own preprocessing logic by porting their existing code into ERASURE. ERASURE's vision is that the data can be partitioned through a cascade of configurable *DataSplitters*. ERASURE already provides the splitters that solve the typical selection scenarios, e.g., by percentages, based on class groupings, or by a fixed sample count. This versatility enables users to define data partitions (such as training, testing, forget, or retain sets) to suit their specific experimental protocols without coding.

Unlearning Methods - The *Unlearner* class encompasses the common logic of each unlearning strategy. In ERASURE, we specifically included the most adopted and widely recognized Unlearners from the literature to evaluate the key dynamics of the machine unlearning field, ranging from retraining-based methods to efficient approximate techniques such as selective gradient dampening and synaptic decay: ***Gold-Model***, ***Fine-Tuning***, ***Successive Random Labels***, ***CF-k*** [Goel *et al.*, 2022], ***EU-k*** [Goel *et al.*, 2022], ***NegGrad*** [Golatkar *et al.*, 2020], ***Advanced NegGrad*** [Choi and Na, 2023], ***UNSIR*** [Tarun *et al.*, 2023], ***Bad Teaching*** [Chundawat *et al.*, 2023a], ***SCRUB*** [Kurmanji *et al.*, 2024], ***Fisher Forgetting*** [Golatkar *et al.*, 2020], ***Selective Synaptic Dampening*** [Foster *et al.*, 2024], and ***Saliency Unlearning*** [Fan *et al.*, 2023]. Moreover, the 13 available Unlearners have been implemented and refactored, with the possibility of chaining (i.e., by adopting the *Cascade* class) them together or applying different combinations of them, e.g., using any unlearning method by exploiting the saliency maps generated by SalUn.

Evaluation - The *Evaluator* module incorporates all the necessary components for assessing the different kinds of performance of unlearning techniques, such as efficiency and efficacy. To facilitate large-scale experiments, the *Evaluator Manager* automates the setup and evaluation process by initializing all the required *Measures* (see its abstract class) and by executing them sequentially. A shared *Evaluation* object collects the results of the evaluated measures. The evaluation starting point (see the *Runners* measure class) might include **efficiency** performance metrics such as the running time, the FLOPS (floating point operations), and memory usage (implemented through PAPI [ICL, 2024] and torch Profiler, because these metrics must be measured during the running of each Unlearning method. With respect to **efficacy**, we implemented UMIA (i.e., Unlearning-specific MIA) [Hayes *et al.*, 2024], Relearning Time [Tarun *et al.*, 2023; Golatkar *et al.*, 2021; Golatkar *et al.*, 2020], Anamnesis Index

Listing 1: Structure of the main configuration snippet for an ERASURE experiment. $\langle NS \rangle$ compacts sub-modules namespace.

```
1  "data": {"class":"<NS>.DatasetManager",
2        "parameters": { "DataSource": {...},
3        "partitions": ["p_1", ..,"p_n"]}},
4  "predictor": {"class": "<NS>.TorchModel",
5   "parameters": {
6     "optimizer": {"class": "torch.*.Adam"},
7     "loss_fn": {"class": "torch.*.
          CrossEntropyLoss"},
8     "model":{"class":"<NS>.ResNet18"}}},
9  "unlearners":[
10    {"class": "<NS>.GoldModel",
          "parameters": { ... }},
11    {"class":"<NS>.AdvancedNegGrad",
12       "parameters": { "optimizer":
             {...},],
13 "evaluator":{"class": "<NS>.Evaluator",
14   "parameters": { "measures":[ ... ] } }
```

Listing 2: Detail of the data management configuration of an ERASURE experiment. It replaces Lines 3–4 of Listing 1.

```
1 "DataSource": {
2  "class":"<NS>.TVDataSourceCelebA",
3  "parameters": {"path":"torchvision.
       datasets.CelebA"},
4  "partitions": [
5     {"class": "<NS>.DataSplitterByZ",
6     "parameters": "parts_names":["forget",
          "other_ids"],
7           "z_labels":[2820, 3227, ... ] },
8     {"class":"<NS>.DataSplitterPercentage",
9     "parameters": {"parts_names":["ret", "
          test"], "percentage": 0.8,
10          "ref_data":"other_ids"}},
11    {"class": "<NS>.DataSplitterConcat",
12    "parameters":{"parts_names":["t", "-"],
13    "concat_splits":["ret", "forget"]}}] }
```

[Chundawat *et al.*, 2023b] and Adaptive Unlearning Score (AUS) [Cotogni *et al.*, 2023]. For **accuracy**, all metrics in scikit-learn [Pedregosa *et al.*, 2011] are available.

3 Proof of Concept

The ERASURE users (i.e., researchers and practitioners) can define their experiment through a main configuration file (in JSON notation), which enables the design and execution of complex testing scenarios in a straightforward and reproducible manner. Listing 1 outlines the structure of the main configuration, which consists of four fundamental sections: *data*, *predictor*, *unlearners*, and *evaluator*. Hereafter, we discuss a specific case study that shows how ERASURE enables an experiment to be made easily and reproducibly.

To do so, we choose to model one of the most referenced but challenging scenarios of the unlearning literature, i.e., we need to unlearn a set of individuals that were part of the training set of a binary classifier able to determine whether the person in the input image is smiling or not. As shown in the main configuration (Listing 1), we defined the *ResNet18* model in the *predictor* JSON object (Line 4) and the *CelebA* dataset [Liu *et al.*, 2015] as the dataset and we indicated the binary attribute "is_smiling" as ground truth. Technically, the dataset and the partitions we want to use are defined in the *data* JSON object (Line 1 of Listing 1) by specifying the *DataSource* and the chain of partitions/subsets obtained through the *Splitters* (see Line 1 of Listing 1 and the details reported in Listing 2). In this case, the CelebA dataset is directly gathered from TorchVision (Line 2) and then divided into four partitions by the following sequence of three splitters: *i)* (Line 5) the *Forget Set* is defined through persons' IDs from the whole dataset; *ii)* (Line 8) the *Retain Set* (80%) and the *Testing Set* (20%) are then obtained from the remaining data ($other\_ids$) of the previous splitter; *iii)* (Line 11) finally, the *Train Set* is obtained by concatenating the Retain and the Forget sets. This sequence of operations is designed to show the potential of ERASURE and guarantee that the Forget Set is not part of the Test Set. The *predictor* JSON object allows to specify all the model details. In this case, we configured the *TorchModel* to directly use the PyTorch networks and set all its parameters, like the optimizer and loss function (notably, in Lines 6 and 7, we use the original implementations and parameters without requiring further modifications). The *unlearners* JSON object (line 9) defines the list of unlearners, which will be evaluated in sequence but in an isolated manner. For example, lines 11-12 of Listing 1 show how to define the *AdvancedNegGrad* unlearner.. The *evaluator* JSON object (13) defines the list of measures that must be used with their parameters – omitted here for the sake of space. Once the experiment is defined, it is possible to run it to produce the results that we reported in Table 1.

Method	Time (s)	Acc	AUS	RT	AIN	UMIA
Gold Model	2048.487	0.921	0.980	2	1.000	0.525
AdvNegGrad	644.660	0.924	0.946	0	0.004	0.544
FineTuning	321.668	0.918	0.969	4	1.995	0.517
CF-k	257.491	0.923	0.949	100	49.756	0.521
EU-k	2632.165	0.923	0.951	35	17.418	0.540
SuccRandLabels	275.832	0.922	0.970	2	1.000	0.525
NegGrad	1.479	0.922	0.959	4	2.000	0.537
Original Model	/	0.924	0.946	0	0	0.544

Table 1: Results of the Proof of Concept. Acc = Accuracy, RT = Relearning Time (in Epochs).

As proof of concept, all baseline Unlearners maintain the same accuracy as the Original Model but differ significantly in Execution Time. NegGrad is the fastest among them, while re-training the model from scratch (Gold Model) is, as expected, among the slowest.

Acknowledgements

The numerical simulations have been realized on the HPC cluster of the Department of Information Engineering, Computer Science and Mathematics (DISIM) at the University of L'Aquila. This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013), and it is partially funded by the National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR) - Project: "SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics" - Prot. IR0000013 - Avviso n. 3264 del 28/12/2021. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Contribution Statement

Andrea D'Angelo and Claudio Savelli contributed equally to this work. Giovanni Stilo is the corresponding author.

References

[Choi and Na, 2023] Dasol Choi and Dongbin Na. Towards machine unlearning benchmarks: Forgetting the personal identities in facial recognition systems. *arXiv preprint arXiv:2311.02240*, 2023.

[Chundawat *et al.*, 2023a] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7210–7217, 2023.

[Chundawat *et al.*, 2023b] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354, 2023.

[Cotogni *et al.*, 2023] Marco Cotogni, Jacopo Bonato, Luigi Sabetta, Francesco Pelosin, and Alessandro Nicolosi. Duck: Distance-based unlearning via centroid kinematics. *arXiv preprint arXiv:2312.02052*, 2023.

[Crawford, 2022] Kate Crawford. *Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence*. Yale University Press, 2022.

[Fan *et al.*, 2023] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*, 2023.

[Foster *et al.*, 2024] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12043–12051, 2024.

[Goel *et al.*, 2022] Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. Towards adversarial evaluations for inexact machine unlearning. *arXiv preprint arXiv:2201.06640*, 2022.

[Golatkar *et al.*, 2020] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9304–9312, 2020.

[Golatkar *et al.*, 2021] Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 792–801, 2021.

[Hayes *et al.*, 2024] Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. Inexact unlearning needs more careful evaluations to avoid a false sense of privacy. *arXiv preprint arXiv:2403.01218*, 2024.

[HuggingFace, 2025] HuggingFace. Hugging face: Natural language processing and machine learning tools. `https://huggingface.co`, 2025. Accessed: 2025-05-01.

[ICL, 2024] ICL. The performance application programming interface (papi). `https://en.wikipedia.org/wiki/Performance_Application_Programming_Interface`, 2024. Accessed: 2025-05-01.

[Kelly *et al.*, 2025] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The uci machine learning repository. `https://archive.ics.uci.edu`, 2025. Acc.: Jan 22, 2025.

[Koudounas *et al.*, 2025] A. Koudounas, C. Savelli, F. Giobergia, and E. Baralis. " alexa, can you forget me?" machine unlearning benchmark in spoken language understanding. *arXiv preprint arXiv:2505.15700*, 2025.

[Kurmanji *et al.*, 2024] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36, 2024.

[Liu *et al.*, 2015] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[Mantelero, 2013] Alessandro Mantelero. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235, 2013.

[Nguyen *et al.*, 2022] Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv:2209.02299*, 2022.

[Pedregosa *et al.*, 2011] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and

E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011.

[Tarun *et al.*, 2023] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[TorchVision, 2025] TorchVision. Torchvision: Pytorch's computer vision library. `https://pytorch.org/vision`, 2025. Accessed: 2025-05-01.