On the Evaluation of Machine Unlearning Methods: A Multi-domain Classification Benchmark

Machine Learning (2026) 115:161
https://doi.org/10.1007/s10994-026-07094-y

On the Evaluation of Machine Unlearning Methods: A Multi-domain Classification Benchmark

Andrea D'Angelo[1,4] · Claudio Savelli[2] · Flavio Giobergia[2] · Elena Baralis[2] · Giovanni Stilo[3]

Received: 16 January 2026 / Revised: 5 May 2026 / Accepted: 1 June 2026 /
Published online: 29 June 2026

Abstract
Machine Unlearning (MU), the process of removing specific data influences from trained machine learning models, is critical for regulatory compliance (e.g., GDPR's right to be forgotten) and for addressing copyright and privacy concerns in large-scale models. While a wide range of methods and metrics have been proposed, systematic evaluations remain fragmented, typically limited in scope by modality, metric coverage, or the number of methods considered. Moreover, the lack of standardized benchmarks leaves several gaps in evaluation protocols, including how to efficiently compare methods, identify optimal hyperparameters, and determine which experimental settings are appropriate for fair and meaningful benchmarking. To address these gaps, we present the most comprehensive MU benchmark to date, evaluating 12 unlearning methods across 8 classification datasets, 4 modalities, several hyperparameters and settings. Based on previous literature and our empirical results, we formalize evaluation protocol desiderata to guide future MU benchmarking. Following these guidelines, we report benchmark results highlighting the best methods within and across domains. To help with method comparison, we also introduce LUMA, a unified metric that aggregates core unlearning dimensions into a single score. Our code is reproducible and extensible to serve as a benchmark for MU research.

Keywords Machine unlearning · Benchmark · Classification · Evaluation protocol

1 Introduction

The growing adoption of machine learning (ML) models across industries has raised concerns about the use of potentially sensitive data in model training (Grynbaum & Mac, 2023). In response, regulations such as the AI Act and the General Data Protection Regulation (GDPR) established the *right to be forgotten*, mandating that an individual's data must be

Andrea D'Angelo and Claudio Savelli have contributed equally to this work.

Editors: Zahraa S. Abdallah, Annalisa Appice, Przemyslaw Biecek, Andrea Tagarelli.

Extended author information available on the last page of the article

removed upon request (Mantelero, 2013). In such cases, the model's owner is expected to release a new version of the model that specifically excludes the targeted samples from the training set. However, retraining ML models from scratch upon every request is often too expensive in terms of time, money, and environmental costs (Crawford, 2022). *Machine Unlearning* (MU) offers a promising alternative: instead of re-training the model from scratch, MU aims to efficiently remove the influence of specific data points from an already trained model (Xu et al., 2024).

Despite its recent emergence, the literature on MU has been growing rapidly, outpacing benchmarking and survey efforts. While methods are often presented as generally applicable (Chundawat et al., 2023; Foster et al., 2024), most benchmarks focus on a single domain, such as images (see Sect. 2), which limits the ability to assess their generalizability. The Tabular domain, in particular, remains largely unexplored; to the best of our knowledge, ours is the first MU benchmark to cover it. For the graph domain, we focus on graph classification, which, in contrast to node- and edge-level classification, has also been unexplored. The textual domain has received more attention; however, existing benchmarks focus on text generation rather than classification (Maini et al., 2024).

Moreover, while some works have raised concerns about the evaluation protocols of MU methods (e.g., the number of seeds (Lanyon et al., 2025) and the evaluation dimensions (Zhao et al., 2024)), there is a critical gap due to the absence of a well-defined, commonly accepted evaluation protocol. Current studies largely adopt ad hoc experimental designs, selecting metrics, hyperparameters, and baselines in an inconsistent and often incomparable manner. Consequently, reported results are difficult to reproduce and compare across works, and conclusions about the relative effectiveness of MU methods frequently depend on arbitrary evaluation choices.

To establish coherence in this fragmented landscape, in this work, we propose three research questions (RQs), and answer each with a novel contribution:

RQ1. How can we enable fast, fair and customizable comparison between different unlearning methods, as well as across hyperparameter settings for a fixed unlearner? To answer RQ1, we introduce LUMA, a unified metric that jointly captures the three key aspects of MU: utility, efficacy, and efficiency. LUMA enables direct comparison across unlearning methods while allowing deployment-specific weighting of evaluation dimensions. We show that it outperforms existing metrics both theoretically and empirically, and adopt it as the core metric in our benchmark.

RQ2. What criteria and evaluation protocols should be used to benchmark a MU method? To answer RQ2, we build our benchmark grounded in a principled review of prior literature and extensive empirical analysis. The first stage of our evaluation focuses on identifying which aspects and hyperparameters of an MU method are meaningful and informative to test. Building on this foundation, we derive actionable guidelines for evaluating and comparing MU methods fairly.

RQ3. To what extent do existing unlearning methods generalize across domains, and are certain approaches consistently superior in specific application settings? Finally, to answer RQ3, we focus on methods intended to generalize across domains. In several cases, we had to adapt their implementations, which constitutes an additional contribution of our benchmark. We build a taxonomy of the benchmarked methods to aid in structuring and understanding the field. Then, we use it to show the results of our benchmark, which is the most comprehensive evaluation of MU methods to date, covering 12 MU methods across 8

models and 8 classification datasets (spanning four domains), including the first benchmarks for tabular data. The publicly available benchmark we provide is reproducible, easily extensible, and intended to serve as a foundation for future research in MU.

As illustrated in Fig. 1, our unlearning benchmark consists of four steps: (1) selecting datasets from one of the four supported domains, (2) training the appropriate model (in two sizes), and (3) applying unlearning methods directly to the model without requiring domain-specific adjustments (using three different forget set sizes). After unlearning, (4) involves collecting and analyzing various evaluation metrics. By proposing this benchmark and leveraging it to answer these three research questions, we aim to standardize the evaluation of MU methods through actionable guidelines and the introduction of LUMA.

The paper is structured as follows. In Sect. 2, we compare with related work, and show how ours is the most comprehensive benchmark to date. In Sect. 3, we show the design decisions, grounded in the literature, that we followed to build our benchmark. Section 4 introduces LUMA, our unified metric, and proves its superiority to existing metrics. Then, Sect. 5 shows results of our benchmark, from which we derive actionable guidelines for evaluating MU methods, and we infer which general methods work best on each domain. Lastly, Sect. 6 concludes the paper.

We ran all experiments in the ERASURE framework (D'angelo et al., 2025a, 2025b). All experiments are reproducible, and the configuration files are available on Github.[1]

[1] https://github.com/aiim-research/ERASURE/tree/main/configs/benchmark.

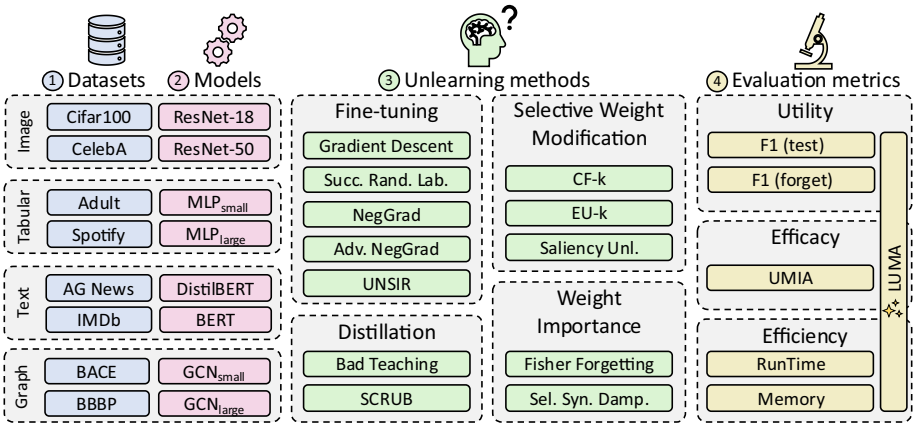

Fig. 1 Experimental workflow of our benchmark. We evaluate 8 classification datasets across 4 domains, training 2 models per domain. We test 12 unlearning methods categorized in 4 groups (based on their unlearning strategy) and assess them across 3 evaluation dimensions. We introduce a unified metric, LUMA, to facilitate comparison

Table 1 Comparison of classification benchmarks present in the literature

	Coverage		Modalities						Metrics		
	Datasets	Methods	Image	Text	Tabular	Graph	Speech	Video	Utility	Efficacy	Efficiency
Our	8	12	✓	✓	✓	✓	✗	✗	✓	✓	✓
Cheng and Amiri (2024)	6	5	✓	✓	✗	✗	✓	✓	✓	✗	✓
Choi and Na (2023)	2	7	✓	✗	✗	✗	✗	✗	✓	✓	✗
Cadet et al. (2024)	5	$11 + 7^2$	✓	✗	✗	✗	✗	✗	✓	✓	✓
Grimes et al. (2024)	1	6	✓	✗	✗	✗	✗	✗	✓	✗	✗
Koudounas et al. (2025)	5	8	✗	✗	✗	✗	✓	✗	✓	✓	✓

[2] 7 of the methods reported in this survey were taken from a Machine Unlearning competition and, as such, were not formally peer-reviewed

2 Related Work

As an emerging field, machine unlearning has yet to undergo systematic evaluation of the extensive research developed over recent years. In addition, most existing benchmarks remain restricted to a single domain. We list these studies in Table 1, providing details on their coverage (datasets and methods), the domains they cover, and the number of MU aspects they capture with their metrics.

Several works focus on a single domain: Grimes et al. (2024); Choi and Na (2023); Cadet et al. (2024) compare unlearning approaches exclusively on image datasets. Similarly, Koudounas et al. (2025) focuses solely on spoken language understanding datasets.

While valuable, none of these methods were tested across multiple domains, an omission that critically limits our understanding of MU methods' generalizability. Moreover, most benchmarks either do not consider all the evaluation dimensions or are very limited in the number of datasets and methods they employ.

To the best of our knowledge, the benchmark introduced by Cheng and Amiri (2024) remains the only existing effort to evaluate MU methods across multiple domains. However, it lacks a crucial component of MU evaluation: metrics that directly assess the efficacy of the unlearning process. In their absence, the reported results lack rigorous quantification of unlearning effectiveness, undermining the ability to address the core privacy and security goals that MU is meant to achieve (Xu et al., 2024). In addition, as shown in Table 1, their study covers a far narrower set of methods (5 versus our 12), which restricts both the scope and generalizability of their findings.

Table 1 shows that no benchmark in the literature has assessed MU methods in the Tabular and Graph domains. Conversely, our benchmark fills this gap by considering 8 datasets (2 per domain, more than any other benchmark) and 12 methods across four different data domains: Image, Text, Tabular, and Graph. In addition, each selected domain is supported by a complete evaluation pipeline that jointly considers utility, efficacy, and efficiency, and systematically varies experimental factors such as forget set construction and model size. This allows to isolate the behavior of unlearning methods across modalities and scenarios while maintaining comparability and reproducibility. Moreover, we incorporate all key dimensions of MU evaluation within a single framework, introducing LUMA (see Sect. 4), the first unified multidimensional metric. At the same time, the benchmark is designed to be extensible: its modular structure enables the straightforward inclusion of additional domains as standardized datasets, models, and evaluation protocols become available.

By providing the most extensive and systematic benchmark to date, we bring coherence to the fragmented landscape of MU in classification tasks and establish an extensible reference point for validating future methods.

3 Benchmark Design

We formalize the standard MU workflow as follows: a model M is first trained on a dataset D. A forget set D_f (i.e., a subset of samples to be removed) is defined upon receiving an unlearning request. As a consequence, the retain set is defined as $D_r = D \setminus D_f$. The goal of an unlearning method is to transform M into an updated model M' that closely approximates a retrained "Gold" model, obtained by retraining from scratch on D_r.

We ground the design of our benchmark in this workflow, as illustrated in Fig. 1. We select datasets, models, hyperparameters, and evaluation metrics based on established practices in the existing literature. In this Section, we describe each of these components in detail and provide the corresponding references.

3.1 Datasets and Models

We evaluate all unlearning methods across four domains (image, text, tabular, and graph) using two publicly available datasets per domain. For domains with established MU benchmarks, we adopt datasets widely used in prior work (Cha et al., 2024; Cheng & Amiri, 2024; Chundawat et al., 2023; Fan et al., 2023; Foster et al., 2024; Goel et al., 2022; Golatkar et al., 2020; Tarun et al., 2023). For domains not explored in MU, we select representative, widely studied datasets (Le Quy et al., 2022; Wu et al., 2018).

The forget set D_f for each dataset is constructed following one of two strategies: (i) selecting training samples containing predefined named entities (e.g., person or organization names) or identity-related attributes **(NE)**, simulating realistic deletion requests; or (ii) sampling a percentage of the training set uniformly at random **(SA)** when identity-based filtering is not feasible.

In particular, for named entities, we select a subset of identities (e.g., individuals in CelebA) and remove all associated samples. This approach requires identity-level annotations for each sample, that is not available for all datasets; although more demanding, it better reflects realistic unlearning scenarios where all data related to a specific entity must be deleted.

In both cases (**NE** and **SA**), three forget set sizes are considered for each dataset: 1%, 5% or 20% of the training data (Hayes et al., 2024; Kurmanji et al., 2024). For NE, we select enough identities to fulfill each threshold. Since the number of samples per identity is small relative to the total dataset size, the resulting deviation from the target fraction is negligible.

For the **image** domain we selected CIFAR-100 (Krizhevsky, 2009) ***(SA)*** and CelebA (Liu et al., 2015) ***(NE)***. CelebA was used for multilabel classification. For **text** we selected IMDB (Giobergia, 2023) ***(NE)*** and AG News (Gulli, 2005) ***(NE)***. For **tabular** data, we selected the datasets of Adult (Becker & Kohavi, 1996) ***(SA)*** and Spotify Tracks (maharshipandya: Spotify Tracks Dataset, 2023) ***(SA)***, and for **graphs** we selected BBBP (Sakiyama et al., 2021) ***(SA)*** and BACE (Wu et al., 2018) ***(SA)***. More details are reported in Section B.1.

For each domain, we adopt model architectures widely used in the MU literature, selecting both smaller and larger variants. In the **image** domain, we use ResNet-18 and ResNet-50 (He et al., 2016), consistent with prior MU studies (Fan et al., 2023; Foster et al., 2024; Golatkar et al., 2020; Tarun et al., 2023). For **text**, we fine-tune DistilBERT (Sanh et al., 2020) and BERT (Devlin et al., 2019). In the **tabular** domain, we employ two fully connected networks with one and three hidden layers, each containing 100 units. Finally, for **graphs**, we use two GCN-based classifiers with a backbone of one or two GCN layers followed by one dense layer.

We trained all models until convergence with early stopping. Full training configurations and hyperparameters are detailed in Appendix B.2.

3.2 Unlearning Methods

For this benchmark, we include 12 different state-of-the-art unlearning methods. Since our goal is to evaluate the generality and cross-domain adaptability of MU methods, we focus on general-purpose unlearning approaches and omit domain-specific methods (e.g., graph MU methods). This choice ensures that all evaluated methods can be applied uniformly across domains and that performance differences reflect unlearning behavior rather than domain-tailored design. While these methods are described as general-purpose in the literature, several were originally proposed and evaluated only within specific domains (most commonly images), and are not always directly applicable across modalities. To ensure a fair cross-domain comparison, we adapt the implementation of such methods when necessary to operate under a unified setting. Among these, UNSIR was originally designed for class-level unlearning in image classification, and we extend it to support entity-level unlearning on text. To facilitate analysis and discussion, we categorize them into four groups based on their unlearning strategy:

Fine Tuning (**FT**): Gradient Descent (`GD`), Successive Random Labels (`SRL`), NegGrad (`NG`) (Golatkar et al., 2020), Advanced NegGrad (`ANG`) (Choi & Na, 2023), UNSIR (`UNSIR`) (Tarun et al., 2023). These methods further train the model using specific strategies.

Selective Weight Modification (**SWM**): CF-k (`CF`k) (Goel et al., 2022), EU-k (`EU`k) (Goel et al., 2022), Saliency Unlearning (`SalUn`) (Fan et al., 2023). These methods operate only on a subset of the model parameters (e.g., the last layers or the weights identified via saliency masking).

Distillation (**DIS**): Bad Teaching (`BT`) (Chundawat et al., 2023), SCRUB (`SCRUB`) (Kurmanji et al., 2024). These methods rely on teacher–student setups to alter the target model's behavior.

Weight Importance (**WI**): Fisher Forgetting (`FF`) (Golatkar et al., 2020), Selective Synaptic Dampening (`SSD`) (Foster et al., 2024). These directly modify model weights, leveraging the Fisher Information Matrix to estimate parameter importance with respect to D_f.

For reference, the metrics related to the ***Original*** (`Orig.`) model and the ***Gold Model*** (`Gold`) are also reported for each dataset. In a few cases, some unlearning methods required minor modifications (e.g., changes to the loss function) to work consistently across settings.

3.3 Hyperparameters

When it comes to unlearning techniques, there is no clear strategy for hyperparameter selection in the literature. All the methods we employ rely on the definition of a learning rate for the unlearning process, so we tune this hyperparameter. We consider the same learning rate value used during the training process, as well as one order of magnitude higher (i.e., an "aggressive" unlearning) and lower (i.e., a "softer" unlearning).

Additionally, we tune method-specific parameters where relevant (e.g., the dampening constant in Selective Synaptic Dampening (Foster et al., 2024). Moreover, as Lanyon et al. (2025) argue, one seed is insufficient to faithfully evaluate MU methods. For this reason, all experiments were run three times with different seeds to account for statistical variation.

3.4 Metrics

We evaluate unlearning methods along three key dimensions (Zhao et al., 2024), which we refer to as *Evaluation Dimensions* (*ED*):

Utility: the predictive performance of the model after unlearning to evaluate the unintended degradation on non-forgotten samples. We compute the F1 score on both the test and the forget sets.

Efficacy: the extent to which the influence of the forget set is removed. We adopt the Unlearning Membership Inference Attack (UMIA) (Hayes et al., 2024), which tests whether an attack model can distinguish forgotten samples from previously unseen ones. Effective unlearning yields UMIA values close to those of the Gold model, avoiding both *over*- and *under*-unlearning (Koudounas et al., 2025).

Efficiency: the computational cost of MU. We measure the time to unlearn relative to full retraining and the peak GPU memory usage during execution.

4 LUMA: A Unified Metric

Although analyzing the three *ED*s separately provides a detailed assessment of the performance of an unlearning method, a unified MU metric is necessary for several reasons:

- It enables straightforward comparison across different methods.
- It allows for immediate identification of the best-performing hyperparameters once a method is fixed, since each configuration is summarized by a single value.
- It offers flexibility for real-world applications: by adjusting the aggregation hyperparameters, it is possible to explicitly express preferences among each *ED*, prioritizing utility over efficacy or efficiency, or vice versa, depending on the specific deployment requirements (Blanco-Justicia et al., 2025).

The need for a single aggregated value is well established in general ML. For instance, the F1 score combines precision and recall through a harmonic mean to capture their trade-off in a single score. Similarly, recent works in MU have highlighted the importance of aggregating multiple evaluation aspects into a unified score, often adopting harmonic means across metrics (Koudounas et al., 2025), like in the case of TOFU for LLM Unlearning (Maini et al., 2024). (Koudounas et al., 2025) introduces GUM, a Global Unlearning Metric, but it faces limitations in considering multiple measures and in resilience to edge cases. GUM reduces each dimension to a single proxy measure (e.g., UMIA for Efficacy, F1 score for Utility, RunTime for Efficiency), which fails to capture the richness of MU performance when multiple measures per *ED* are available: in practice, MU performance is evaluated by several measures (e.g., F1 scores on forget and test sets, RunTime and GPU memory usage), and considering all these aspects jointly provides a more faithful assessment than relying on single-value proxies. (Zhao et al., 2024) proposed Tug of War (ToW), defined as the product of the relative differences between the Gold and unlearned models on accuracies over the test, retain, and forget sets. However, ToW excludes a direct metric for assessing Efficiency, resulting in a partial evaluation. We provide empirical evidence of these shortcomings in Appendix A.1.

To address these limitations, we introduce the **Laplacian Unlearning Multidimensional Assessment (LUMA)**. Unlike GUM and ToW, LUMA is multidimensional by design, incorporating vectors of measures within each *ED*, and flexible, allowing users to add any measure deemed relevant. LUMA computes the similarity of an unlearned model to the retrained Gold model across utility, efficacy, and efficiency, while penalizing large deviations in any single dimension.

In our benchmark, the efficacy dimension is measured by UMIA, the utility dimension by F1 score on the test and forget sets, and the Efficiency dimension by RunTime and peak GPU memory usage. Each *ED* can therefore be represented as a vector of the values computed by the chosen measures on the considered model $\cdot \in \{\text{Gold}, M'\}$

$$\text{e.}_{\cdot} = [\,\text{UMIA.}_{\cdot}\,]^{\top},\ \text{u.}_{\cdot} = \left[F1_{\cdot}^{test},\ F1_{\cdot}^{forget}\right]^{\top},\ \text{t.}_{\cdot} = \left[\text{RunTime.}_{\cdot},\ \text{Memory.}_{\cdot}\right]^{\top}$$

All *ED*s are referenced against the Gold Model (`Gold`), which represents the ideal target of MU methods (Xu et al., 2024) as the model retrained from scratch only on the retain set. To compare the unlearning method against the Gold Model, we map efficacy and utility into similarity factors using a γ-parametrized Laplacian kernel. This choice penalizes *under*- or *over*-performance relative to the Gold Model, while amplifying large deviations in any individual measure.

$$M_U(\text{u}_{\text{Gold}}, \text{u}_{M'}) = \exp((-\gamma \|\text{u}_{\text{Gold}} - \text{u}_{M'}\|_1))\,,$$
$$M_E(\text{e}_{\text{Gold}}, \text{e}_{M'}) = \exp((-\gamma \|\text{e}_{\text{Gold}} - \text{e}_{M'}\|_1))\,.$$

While their closeness to the Gold Model evaluates utility and efficacy, efficiency follows a different principle: the less time and memory a method consumes, the better; the Gold Model provides an upper bound reference point. To capture this, we first normalize each efficiency measure by the corresponding value of the Gold Model. We then apply a logarithmic transformation to smooth extreme differences and emphasize relative improvements over absolute ones. Finally, we apply an exponential decay so that larger deviations from the Gold baseline are penalized more strongly, and aggregate the resulting values through a weighted average to obtain a single score:

$$M_T(\text{t}_{\text{Gold}}, \text{t}_{M'}) = \sum_i \text{w}_i \exp\left[-\left(\frac{\log(1+\text{t}_{M',i})}{\log(1+\text{t}_{\text{Gold},i})}\right)^3\right],$$

where w_i represents the weight assigned to the i^{th} efficiency measure. Finally, we define the unified metric as:

Definition 1 *(LUMA)* Let M_U, M_E, M_T be the similarity scores for utility, efficacy, and efficiency relative to the Gold model (as defined above). Then

$$\text{LUMA} = \frac{3}{\frac{1}{M_U} + \frac{1}{M_E} + \frac{1}{M_T}}.$$

In other words, LUMA is the harmonic mean of M_U, M_E, and M_T. LUMA is parametrized by the γ of the Laplacian kernels and the vector w of non-negative weights summing to 1, assigned to the efficiency measures. In our setting, we set $\gamma = 3$ and $\mathrm{w} = (0.5, 0.5)$, simply assigning 50% of the weight to RunTime and 50% to memory efficiency. We detail parameter tuning of LUMA in Sect. A.3.

Note that, due to LUMA 's definition, the Gold Model is always given the same LUMA score in a fixed experimental setting. This is because $M_U = M_E = 1$ for the Gold Model, while M_T depends only on the chosen efficiency measures and weights. Therefore, once the benchmark setting is fixed, the Gold score is constant and serves as a stable reference point against which all unlearning methods are compared. With the hyperparameters set in this study, LUMA is always 0.636 for the Gold Model. We refer the reader to Appendix A.3 for more information. Consequently, any unlearning method that takes longer than the Gold Model will inevitably be penalized by receiving a lower LUMA score, even if it achieves perfect similarity with the Gold Model. This property makes it easy to identify cases where an unlearning method is not suitable.

LUMA offers several advantages: **(i).** It is designed to range from 0 to 1, where 1 is the ideal MU algorithm that returns a model identical to the Gold Model with no additional cost in time or memory usage. **(ii).** It strongly penalizes any significant deviation on any measure, enabling fast pruning of ill-defined (or ill-parameterized) unlearning methods. **(iii).** It is extensible, allowing the integration of any task-specific measure. For example, ROC can replace F1 in the Utility *ED*, or be added alongside it without further modifications. This flexibility is unique to LUMA and ensures seamless integration with future evaluation protocols. We note that alternative multi-objective evaluation approaches, such as Pareto front analysis, characterize trade-offs across evaluation dimensions. However, such approaches are less suited for large-scale benchmarking settings, where many methods, datasets, and hyperparameter configurations must be compared. In these cases, Pareto-based analysis yields sets of non-dominated solutions rather than a single ranking, making comparison and model selection less straightforward. Metrics such as hypervolume can further aggregate these trade-offs into a scalar value, but they require additional design choices and can be sensitive to scaling across objectives. For this reason, in our benchmark, we report both the individual evaluation dimensions and a unified scalar metric, enabling inspection along each axis while still providing a single value for comparison. In this context, LUMA can be seen as a principled scalarization of the multi-objective problem, enabling direct ranking, statistical comparison, and efficient hyperparameter selection.

Answer to RQ1. How can we enable fast, fair and customizable comparison between different unlearning methods, as well as across hyperparameter settings for a fixed unlearner?
We introduce LUMA, a unified evaluation metric that improves upon existing metrics in the machine unlearning literature in terms of completeness, robustness, and extensibility. LUMA is designed to accommodate an arbitrary number of evaluation measures grouped along the dimensions of efficacy, utility, and efficiency, and can be adjusted to reflect deployment-specific necessities. By aggregating these dimensions into a single scalar value per unlearning method, LUMA enables direct comparison across methods and substantially simplifies hyperparameter selection and pruning.

5 Benchmark Evaluation

In this Section, we summarize the main findings from our benchmark study. Specifically, in Sect. 5.1, we leverage the results of our benchmark to extract actionable guidelines for evaluation of MU methods. Then, following these guidelines, in Sect. 5.2 we report the main results from our experimentation on the four domains.

Experiments were conducted on a Linux-based (Centos 8) high-performance computing cluster equipped with Intel Xeon Gold 6140 M CPUs (2.30 GHz, 108 cores), 880 GB of RAM, and an NVIDIA A100 GPU (80 GB memory).

5.1 Hyperparameter Tuning and Model Selection

When benchmarking MU methods, several design choices and hyperparameters must be carefully considered to ensure a fair comparison. To derive evaluation guidelines, we first identify the key design choices that need to be specified.

(i) Hyperparameters. Most MU methods are only parametrized by their Learning Rate (LR) of either the fine-tuning to be applied, the distillation, or any semi-Newton step they employ (refer to Sect. 3.2 for a taxonomy). While the optimal choice of different hyperparameters cannot be determined a priori, the fact that nearly all approaches rely on a learning rate allows us to derive practical evaluation guidelines. In particular, we can identify reasonable ranges of LR values that should be explored to ensure a fair and meaningful comparison across methods. The only unlearning methods that take as input a parameter different from the LR are the ones in the group of Weight Importance (WI): Fisher Forgetting (`FF`) and Selective Synaptic Dampening (`SSD`). In this case, the tuning was applied to their respective parameters, and its results are reported in the supplementary material. In our benchmark, we tuned these parameters by employing LRs one order of magnitude lower, one order of magnitude higher, and equal to the training LR for each domain.

(ii) Forget set Size. The size of the forget set refers to the percentage of the training set to be considered for removal (Xu et al., 2024). To fully simulate real unlearning scenarios, it should be sampled from Named Entities whenever possible (see Sect. 3.1), a practice that is largely neglected in existing studies. Moreover, there is currently no standard in the literature regarding an appropriate choice for the forget set size. In our benchmark, we study forget set sizes of 1%, 5%, and 20% w.r.t. the training set, covering a range from small to large-scale forgetting (Fan et al., 2024).

(iii) Model Size. The literature does not clearly establish how model size affects the preservation of utility and the effectiveness of unlearning. Moreover, it is unclear if conclusions that are drawn on smaller models can be generalized to larger ones. To fill this gap, as discussed in Sect. 3.1, in our benchmark, we use two models with the same architecture but different sizes (one smaller, one larger), for each domain.

(iv) Number of seeds. As general evaluation practice, experiments should be repeated using multiple random seeds, yet this is surprisingly uncommon in the MU literature (Lanyon et al., 2025). This is particularly important for MU, as many methods are inherently stochastic or rely on randomized optimization procedures (e.g., Stochastic Gradient Descent for fine-tuning methods). Using multiple seeds helps account for variability introduced by randomness and yields more reliable, reproducible results. In our benchmark, we report results averaged over three different seeds for each experiment.

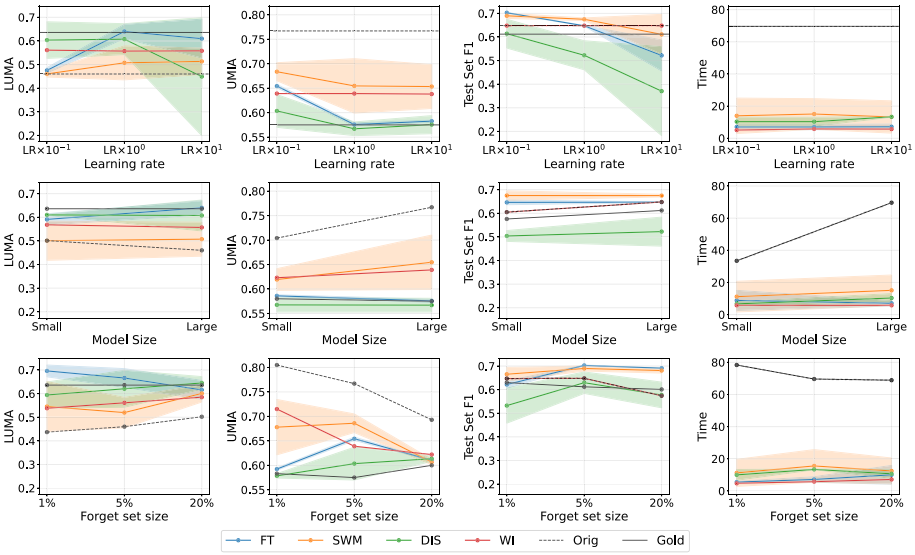

Fig. 2 Sensitivity analysis of unlearning performance across key experimental factors on the CelebA dataset. The figure reports the variation of LUMA, UMIA, test F1 score, and runtime as a function of (top row) the learning rate of the unlearning methods, (middle row) the model dimension, and (bottom row) the forget set size. Each column corresponds to a different evaluation metric, while curves represent different unlearning strategies aggregated by families, together with the original and Gold models

As part of our benchmarking effort, we tested several settings across all four dimensions and show results in Fig. 2. This figure shows results on the CelebA dataset, but the same conclusions hold across datasets and domains. Similar visualizations for other datasets are reported in the Appendix C.1.

Figure 2 shows LUMA, UMIA, F1 score on the test set, and running time for all MU methods. Results are grouped by unlearning strategy (see Sect. 3.2) and plotted against the learning rate scale of the MU method (or the relevant DIS hyperparameter), model size, and forget set size. The learning rate scale k is the scaling factor applied to the learning rate adopted for a specific model. For instance, the learning rate scale $k = 10^1$ applied to a model trained with $LR = 10^{-4}$, implies that the unlearning has been applied with a learning rate of $k \cdot LR = 10^{-3}$.

The first row of Fig. 2 shows that most MU methods, specifically those in the FT and DIS groups, cause sharp drops in utility (F1 score) when the learning rate is too high. This is because the MU method with a higher learning rate is destroying the model's learned representations more aggressively. Although UMIA scores drop closer to 0.5 (the ideal value) and time remains constant, the catastrophic utility loss is enough to make the model unusable (and, consequently, sharply drop LUMA). We infer that the hyperparameter space worth investigating for LR-based MU methods lies around the model's learning rate or one order of magnitude lower.

The second row of Fig. 2 compares the metrics against model size. Results here show a distinct pattern: larger models are less resilient to Membership Inference Attacks after unlearning, less sensitive to utility drops, and MU methods generally take a bit longer. The last point is especially important given the rationale for the MU task: if methods do not save

considerable computational time relative to retraining the model, the latter will always be the best choice. The LUMA scores are also slightly higher for larger models, consistently for all MU methods. We infer that MU methods should be tested on larger models, as they prove to be the hardest settings, and that results from smaller models are not directly generalizable to larger models.

Lastly, the third row of Fig. 2 compares metrics across forget set sizes: 1%, 5%, and 20% of the original training set. As expected, a forget set size of 20% induces utility loss even in the Gold model, but also collapses all UMIA scores closer. Other times (see Appendix C.1, like for CIFAR-100, the UMIA scores are incredibly high when the forget set is Large). In general, the model's behavior when the forget set is too big is largely unpredictable and unstable. Forget sets of 1% (small) or 5% (medium) of the training set are more realistic and stable; we find that medium forget sets are a reasonable middle ground for evaluation.

Answer to RQ2. What criteria and evaluation protocols should be used to benchmark a MU method?
We ground our benchmark design in prior literature and empirical evaluation, and formulate the following guidelines for fair MU evaluation:
(i). Unlearning methods must be evaluated along three complementary dimensions, efficacy, utility, and efficiency, to provide a meaningful assessment [9].
(ii). Following the recommendation of Lanyon et al. [8], we perform unlearning on multiple independently trained models (3 instances per experimental scenario, each initialized with a different random seed) to ensure a reliable evaluation.
(iii). We find that, for learning-rate-based MU methods, it is sufficient to explore learning rates centered around that of the original model, including values up to one order of magnitude lower. In contrast, increasing the learning rate by one order of magnitude is enough to severely degrade model utility.
(iv). The forget set should be around 5% of the original training set. While some prior work considers larger fractions (e.g., 20% or above) [19, 38–40], such settings tend to weaken the model against MIAs. A smaller forget set, constructed around named entities when possible, better aligns with realistic unlearning use cases.
(v). Finally, our results suggest that results obtained on smaller models cannot be generalized to larger ones, even when the underlying architecture is preserved. Larger models are more resilient to UMIA, and take longer to train, which is a key aspect of MU. We recommend using large enough models for MU evaluation.

As a result, in the remainder of the paper we will report only results following these guidelines. Specifically, we will only report results using the same LR as the model's hyperparameter, a forget set of size 5%, and larger models. For completeness, we report all other results in the supplementary material.

5.2 Main Results

In this Section we report the main results of our benchmark. Following Sect. 5.1 guidelines, we report results only for hyperparameters equal to those of the model, a forget set of size 5%, and the largest available model. While LUMA is computed using the full set of measures described in Sect. 3, for the sake of space we only report a subset of representative measures per *ED*, together with the aggregated LUMA for all experiments. However, full results, including runs for smaller models and different parameters, are reported in the

supplementary material. From these tables, we draw intra- (Sect. 5.2.1) and inter-domain (Sect. 5.2.2) observations. Intra-domain observations rely on the mean and standard deviations obtained across 3 runs of our experiments. We corroborate inter-domain observations with statistical tests shown in Sect. 5.2.3.

5.2.1 Intra-domain Observations

Image Domain. Table 2 shows that the best-performing unlearning methods in the image domain, according to LUMA, are GD, SRL, and BT, which achieve comparable results. This holds across both datasets, despite their different setups (CIFAR-100 for multiclass and CelebA for multilabel classification), indicating a degree of stability in image-domain unlearning. This is not surprising, as the image domain is by far the most extensively studied (see Sect. 2). All these methods are able to maintain model utility both on the test set and forget set with respect to Orig., while requiring a fraction of the time of retraining (see

Table 2 Methods comparison (mean ± std) on CIFAR100 and CelebA

	Alg.	LUMA ↑	UMIA →	$F1_t$ →	$F1_f$ →	Time ↓	Mem. ↓
CIFAR-100							
–	Orig.	.549 ± .038	.574 ± .031	.355 ± .007	.446 ± .041	24.48 ± .66	17.28 ± .00
–	Gold	.636 ± .000	.507 ± .011	.332 ± .009	.329 ± .013	24.48 ± .66	17.28 ± .00
FT	GD	.715 ± .036	.556 ± .025	.339 ± .008	.399 ± .018	1.70 ± .05	19.42 ± .00
	SRL	.719 ± .035	.550 ± .020	.337 ± .009	.386 ± .024	1.85 ± .08	22.49 ± .00
	NG	.522 ± .080	.513 ± .017	.136 ± .032	.141 ± .043	**.13** ± .01	26.96 ± 5.08
	ANG	.289 ± .015	**.500** ± .000	.000 ± .000	.000 ± .000	4.02 ± .15	33.12 ± 5.08
	UNSIR	.665 ± .029	.549 ± .021	.351 ± .008	.412 ± .024	2.56 ± .06	42.52 ± 5.05
SWM	CF-*k*	.698 ± .052	.571 ± .030	.337 ± .003	.425 ± .043	1.37 ± .14	18.21 ± 6.28
	EU-*k*	.592 ± .028	.567 ± .025	.341 ± .006	.431 ± .041	13.80 ± 1.05	19.26 ± 5.74
	SalUn	.682 ± .035	.555 ± .027	.342 ± .006	.407 ± .046	5.77 ± 6.22	14.23 ± 8.32
DIS	BT	**.736** ± .024	.536 ± .012	**.334** ± .017	**.363** ± .016	4.12 ± .04	**8.24** ± .00
	SCRUB	.642 ± .045	.561 ± .022	.364 ± .019	.445 ± .047	2.27 ± .09	34.37 ± 15.50
WI	FF	.307 ± .246	.026 ± .889	.107 ± .093	.126 ± .111	38.72 ± 1.35	31.66 ± 17.06
	SSD	.651 ± .043	.566 ± .029	.355 ± .007	.446 ± .041	1.77 ± .07	37.55 ± 15.48
CelebA							
–	Orig.	.460 ± .056	.767 ± .008	.648 ± .029	.803 ± .063	69.62 ± .48	8.32 ± .81
–	Gold	.636 ± .000	.575 ± .002	.612 ± .022	.608 ± .007	69.62 ± .48	8.32 ± .81
FT	GD	.709 ± .041	.581 ± .001	**.587** ± .006	.581 ± .011	6.98 ± 2.79	15.07 ± .02
	SRL	.676 ± .045	.579 ± .004	.646 ± .011	.716 ± .024	5.11 ± .12	**1.47** ± .82
	NG	.207 ± .091	**.575** ± .004	.171 ± .064	.208 ± .077	**.60** ± .04	23.30 ± .01
	ANG	.089 ± .007	.596 ± .006	.043 ± .000	.039 ± .000	13.12 ± .12	3.54 ± .02
	UNSIR	.613 ± .074	.581 ± .004	.547 ± .064	.537 ± .064	9.30 ± .14	36.92 ± .02
SWM	CF-*k*	.495 ± .024	.686 ± .002	.671 ± .006	.871 ± .012	6.60 ± 1.94	16.20 ± .96
	EU-*k*	.456 ± .023	.684 ± .002	.671 ± .006	.870 ± .013	27.45 ± .41	17.30 ± .48
	SalUn	.608 ± .010	.576 ± .003	.667 ± .010	.771 ± .012	5.70 ± .08	15.94 ± 4.09
DIS	BT	.699 ± .007	.557 ± .002	.559 ± .007	**.589** ± .044	13.03 ± .11	6.05 ± .00
	SCRUB	.542 ± .003	.581 ± .017	.461 ± .024	.481 ± .033	7.56 ± 1.03	29.05 ± 13.10
WI	FF	**.759** ± .019	.590 ± .025	.545 ± .023	.654 ± .048	3.46 ± .04	1.84 ± .00
	SSD	.561 ± .079	.639 ± .042	.648 ± .029	.803 ± .063	5.09 ± .82	33.40 ± 13.10

Time is measured in minutes and memory in GB. Best results are highlighted in **bold**: highest for LUMA, lowest for Time and Memory, and closest to the Gold model for UMIA and F1 scores

Gold). However, UMIA is slightly increased after the application of those methods, which means some information is still leaking from the unlearned models. Notably, for CelebA, SSD actually achieves the best utility retention on both sets, but its UMIA is very high, meaning that the resulting model is very vulnerable: for this reason, the LUMA score is lower than the Gold threshold.

Tabular Domain. Table 3 shows that NG is the clear winner on the Adult dataset, followed by GD and FF. These methods also achieve strong performance on the Spotify dataset, although FF loses a bit of model utility compared to Gold. In general, fine-tuning FT methods perform very well in this domain, due to the simpler architecture of MLPs and the extremely fast execution time. This is by far the simplest setting, and LUMA scores are therefore high across the board. However, DIS methods seem not to perform too well, mostly because they take just as long as retraining.

Graph Domain. FT Unlearning methods generally perform best, as shown in Table 4. All methods exhibit extremely low running times and relatively minor deviations from the

Table 3 Methods comparison (mean ± std) on Adult and Spotify

	Alg.	LUMA ↑	UMIA →	$F1_t$ →	$F1_f$ →	Time ↓	Mem. ↓
Adult							
–	Orig.	.633 ± .002	.497 ± .002	.790 ± .001	.799 ± .004	2.08 ± .01	.03 ± .00
–	Gold	.636 ± .000	.499 ± .002	.790 ± .002	.797 ± .001	2.08 ± .01	.03 ± .00
FT	GD	.693 ± .002	.498 ± .001	**.790** ± .001	.802 ± .005	1.12 ± .01	.03 ± .00
	SRL	.672 ± .003	.498 ± .002	.789 ± .001	.800 ± .004	1.18 ± .00	.04 ± .00
	NG	**.823** ± .002	**.499** ± .002	**.790** ± .001	**.798** ± .003	**.06** ± .00	.05 ± .00
	ANG	.554 ± .002	.497 ± .003	.792 ± .000	.800 ± .004	2.21 ± .02	.05 ± .00
	UNSIR	.623 ± .002	.501 ± .002	**.790** ± .002	.803 ± .005	1.24 ± .00	.05 ± .00
SWM	CF-k	.669 ± .001	.500 ± .002	.788 ± .001	.799 ± .006	1.11 ± .00	.04 ± .00
	EU-k	.408 ± .001	.498 ± .003	.789 ± .000	.799 ± .005	1.98 ± .04	.04 ± .00
	SalUn	.607 ± .001	.498 ± .002	**.790** ± .002	**.798** ± .004	1.10 ± .01	.07 ± .00
DIS	BT	.689 ± .003	.496 ± .001	**.790** ± .000	.803 ± .004	2.38 ± .03	.02 ± .00
	SCRUB	.621 ± .004	**.499** ± .003	.793 ± .001	.800 ± .006	1.10 ± .01	.06 ± .00
WI	FF	.776 ± .031	**.499** ± .001	.776 ± .021	.778 ± .024	.08 ± .00	.06 ± .00
	SSD	.613 ± .002	.498 ± .000	**.790** ± .001	.799 ± .004	1.10 ± .00	.06 ± .00
Spotify							
–	Orig.	.603 ± .008	.516 ± .003	.652 ± .008	.680 ± .010	1.68 ± .01	.03 ± .00
–	Gold	.636 ± .000	.494 ± .004	.652 ± .002	.629 ± .011	1.68 ± .01	.03 ± .00
FT	GD	.804 ± .013	.509 ± .008	.644 ± .007	.670 ± .003	.08 ± .00	.03 ± .00
	SRL	.799 ± .016	.501 ± .008	.649 ± .007	.669 ± .007	.09 ± .00	.03 ± .00
	NG	.799 ± .022	.500 ± .014	.648 ± .005	.672 ± .013	**.00** ± .00	.04 ± .00
	ANG	.761 ± .011	.497 ± .009	.646 ± .005	.670 ± .010	.16 ± .00	.04 ± .00
	UNSIR	.768 ± .010	.498 ± .008	.647 ± .011	.674 ± .005	.09 ± .00	.04 ± .00
SWM	CF-k	.777 ± .010	.514 ± .009	.654 ± .005	.685 ± .004	.08 ± .00	.04 ± .00
	EU-k	.657 ± .003	**.493** ± .007	.621 ± .009	.611 ± .004	.82 ± .01	.04 ± .00
	SalUn	.752 ± .003	.504 ± .003	**.652** ± .008	.674 ± .008	.09 ± .00	.06 ± .00
DIS	BT	**.830** ± .011	.498 ± .005	**.652** ± .011	.665 ± .016	.17 ± .01	**.02** ± .00
	SCRUB	.758 ± .018	.507 ± .008	.629 ± .008	.660 ± .003	.09 ± .00	.05 ± .00
WI	FF	.708 ± .016	.503 ± .009	.620 ± .016	**.630** ± .019	.29 ± .00	.06 ± .00
	SSD	.748 ± .014	.502 ± .016	**.652** ± .008	.680 ± .010	.09 ± .00	.06 ± .00

Time is measured in minutes and memory in GB. Best results are highlighted in **bold**: highest for LUMA, lowest for Time and Memory, and closest to the Gold model for UMIA and F1 scores

Table 4 Methods comparison (mean ± std) on BACE and BBBP

	Alg.	LUMA ↑	UMIA →	$F1_t$ →	$F1_f$ →	Time ↓	Mem. ↓
BACE							
–	Orig.	.623 ± .008	.530 ± .005	.355 ± .000	.320 ± .009	1.55 ± .62	.02 ± .00
–	Gold	.636 ± .000	.518 ± .036	.355 ± .000	.320 ± .009	1.55 ± .62	.02 ± .00
FT	GD	.847 ± .008	**.519** ± .018	.355 ± .000	.320 ± .009	.03 ± .01	.02 ± .00
	SRL	**.860** ± .007	.516 ± .043	.355 ± .000	.320 ± .009	.04 ± .01	.02 ± .00
	NG	.854 ± .013	.538 ± .051	.355 ± .000	.320 ± .009	**.00** ± .00	.02 ± .00
	ANG	.852 ± .006	.538 ± .035	.355 ± .000	.320 ± .009	.02 ± .01	.02 ± .00
	UNSIR	.841 ± .023	.546 ± .020	.355 ± .000	.320 ± .009	.04 ± .01	.02 ± .00
SWM	CF-*k*	.847 ± .011	.543 ± .032	.355 ± .000	.320 ± .009	.04 ± .02	.02 ± .00
	EU-*k*	.610 ± .006	.535 ± .015	.355 ± .000	.320 ± .009	1.60 ± .64	.02 ± .00
	SalUn	.836 ± .014	.525 ± .006	.355 ± .000	.320 ± .009	.05 ± .02	.02 ± .00
DIS	BT	.847 ± .017	.529 ± .022	.355 ± .000	.320 ± .009	.06 ± .03	.02 ± .00
	SCRUB	.824 ± .020	.524 ± .020	.355 ± .000	.320 ± .009	.07 ± .03	.02 ± .00
WI	FF	.788 ± .017	.529 ± .025	.355 ± .000	.320 ± .009	.21 ± .01	.02 ± .00
	SSD	.831 ± .018	.555 ± .033	.355 ± .000	.320 ± .009	.05 ± .02	.02 ± .00
BBBP							
–	Orig.	.612 ± .013	.496 ± .002	.746 ± .009	.763 ± .007	1.63 ± .01	.02 ± .00
–	Gold	.636 ± .000	.486 ± .005	.725 ± .022	.735 ± .008	1.63 ± .01	.02 ± .00
FT	GD	.816 ± .025	**.490** ± .010	.745 ± .017	.763 ± .007	.03 ± .00	.02 ± .00
	SRL	**.818** ± .006	.493 ± .014	.735 ± .003	.758 ± .020	.04 ± .00	.02 ± .00
	NG	.769 ± .089	.481 ± .013	.687 ± .090	.699 ± .059	**.00** ± .00	.02 ± .00
	ANG	.797 ± .017	.507 ± .010	**.731** ± .024	.773 ± .016	.02 ± .00	.02 ± .00
	UNSIR	.803 ± .021	.495 ± .012	.745 ± .017	.763 ± .007	.03 ± .00	.02 ± .00
SWM	CF-*k*	.811 ± .022	.492 ± .006	.745 ± .017	.763 ± .007	.03 ± .00	.02 ± .00
	EU-*k*	.609 ± .038	.492 ± .008	.748 ± .009	.771 ± .008	1.46 ± .40	.02 ± .00
	SalUn	.793 ± .028	.493 ± .008	.745 ± .020	.758 ± .009	.04 ± .00	.02 ± .00
DIS	BT	.799 ± .020	.495 ± .005	.743 ± .019	.763 ± .007	.13 ± .05	.02 ± .00
	SCRUB	.793 ± .031	.502 ± .010	.745 ± .017	.763 ± .007	.05 ± .00	.02 ± .00
WI	FF	.758 ± .011	.492 ± .002	.696 ± .027	**.737** ± .036	.20 ± .01	.02 ± .00
	SSD	.809 ± .026	.495 ± .002	.746 ± .009	.763 ± .007	.06 ± .03	.02 ± .00

Time is measured in minutes and memory in GB. Best results are highlighted in **bold**: highest for LUMA, lowest for Time and Memory, and closest to the Gold model for UMIA and F1 scores

F1 test, as GCNs generalize well even with fewer samples, so they aren't impacted as much by the removal of the forget set, and the LUMA scores are very high across the board. This holds across both datasets, although it also highlights the difficulty of completely erasing information from the Gold model itself. Overall, graph unlearning remains underexplored and warrants further investigation, as most works in the literature have only focused on node- and edge-level classification.

Textual Domain. SWM methods are relatively strong, while FF and NG methods collapse, as reported in Table 5. This is because BERT (and LLMs in general) usually finetune the last classification layer, while the bulk of knowledge is encoded in the deeper transformer architecture. As such, methods that selectively modify weights (CF-*k*, EU-*k*, SalUn) remove task-specific information without harming the general representations captured by the transformer layers. Interestingly, WI methods are unable to leverage the Fisher Information Matrix to correctly identify which weights to modify on LLMs, as shown by the low LUMA

Table 5 Methods comparison (mean ± std) on AG News and IMDB

	Alg.	LUMA ↑	UMIA →	$F1_t$ →	$F1_f$ →	Time ↓	Mem. ↓
AG News							
–	Orig.	.472 ± .038	.634 ± .031	.876 ± .017	.902 ± .016	86.1 ± .94	7.8 ± 3.3
–	Gold	.636 ± .000	.536 ± .028	.903 ± .003	.651 ± .045	86.1 ± .94	7.8 ± 3.3
FT	GD	.581 ± .053	.559 ± .007	.913 ± .001	.876 ± .017	28.1 ± .03	31 ± .00
	SRL	.593 ± .050	**.545** ± .006	.913 ± .001	.866 ± .018	29.1 ± .01	**25** ± .00
	NG	.289 ± .040	.852 ± .000	.570 ± .004	.369 ± .040	1.09 ± .00	27 ± .00
	ANG	.339 ± .014	.975 ± .001	.893 ± .002	.316 ± .010	56.9 ± .14	33 ± .00
	UNSIR	.593 ± .065	.554 ± .015	.910 ± .002	.832 ± .050	3.2 ± .01	53 ± 3.6
SWM	CF-*k*	.597 ± .050	.555 ± .017	**.905** ± .005	.873 ± .013	9.98 ± .10	73 ± 4.4
	EU-*k*	.559 ± .040	.556 ± .014	.907 ± .004	.876 ± .007	29.8 ± .29	73 ± 4.4
	SalUn	.578 ± .046	.548 ± .007	.913 ± .000	.864 ± .016	36.3 ± 1.2	35 ± .00
DIS	BT	.181 ± .053	.784 ± .140	.289 ± .044	.421 ± .088	49.8 ± 1.30	31 ± .00
	SCRUB	**.606** ± .060	.563 ± .003	.888 ± .011	**.779** ± .073	37.6 ± .22	60 ± 6.5
WI	FF	.044 ± .005	.855 ± .006	.156 ± .048	.011 ± .001	**.01** ± .00	75 ± 1.6
	SSD	.514 ± .051	.654 ± .012	.879 ± .010	.907 ± .021	29.9 ± .77	33 ± .00
IMDB							
–	Orig.	.588 ± .004	.548 ± .003	.938 ± .004	.997 ± .001	93.6 ± 4.4	43 ± 2.5
–	Gold	.636 ± .000	.501 ± .000	.940 ± .005	.930 ± .007	93.6 ± 4.4	43 ± 2.5
FT	GD	.665 ± .019	.498 ± .000	.893 ± .000	.891 ± .000	22.1 ± .00	**47** ± .00
	SRL	.649 ± .025	.542 ± .003	.939 ± .000	.994 ± .002	23.4 ± .00	55 ± 3.8
	NG	.138 ± .046	.497 ± .000	.429 ± .074	.444 ± .078	**1.28** ± .07	49 ± 5.5
	ANG	.643 ± .028	.504 ± .008	.935 ± .003	.956 ± .011	45.1 ± .11	59 ± 3.8
	UNSIR	.639 ± .022	.539 ± .004	.931 ± .008	.992 ± .001	24.7 ± .08	63 ± 3.8
SWM	CF-*k*	**.693** ± .021	.549 ± .006	**.940** ± .004	.998 ± .001	8.53 ± .04	50 ± 4.8
	EU-*k*	.666 ± .027	.546 ± .000	.941 ± .000	.999 ± .000	16.9 ± .00	**47** ± .00
	SalUn	.627 ± .032	.556 ± .000	.944 ± .000	1.00 ± .000	33.1 ± .00	48 ± .00
DIS	BT	.614 ± .031	.540 ± .000	.890 ± .000	**.948** ± .000	4.45 ± .00	66 ± .00
	SCRUB	.630 ± .025	.542 ± .001	.937 ± .007	.994 ± .005	31.2 ± 1.8	63 ± 13
WI	FF	.184 ± .000	**.499** ± .002	.502 ± .009	.498 ± .002	1.47 ± .04	58 ± 13
	SSD	.637 ± .030	.552 ± .012	.941 ± .000	.998 ± .001	24.5 ± 1.1	61 ± 13

Time is measured in minutes and memory in GB. Best results are highlighted in **bold**: highest for LUMA, lowest for Time and Memory, and closest to the Gold model for UMIA and F1 scores

score for FF and SSD on AG news. Lastly, FT and DIS methods achieve good performance (with a few exceptions) but are held back by the non-trivial cost of further training a model of this size in its entirety.

5.2.2 Inter-domain Observations

The main result of the benchmark is that simple fine-tuning FT methods are the most reliable across the board, reflecting both the early stage of MU research and the ongoing search for more reliable methods. Still, some consistent patterns can be observed across domains. Fine-tuning is often a safe strategy, so FT methods are generally the most reliable, with the notable exception of ANG, which often performs worst in terms of LUMA (claim **a**, refer to Sect. 5.2.3 for a deeper analysis). Because the forget sets we define are heterogeneous (spanning multiple classes), a single epoch of advanced negative gradient updates often leads to

model collapse, resulting in poor F1 scores. Similarly, SWM methods perform consistently well, but they rarely achieve the best performance. Among these, EU-k is consistently the worst in terms of LUMA (claim **b**, that we prove statistically in Sect. 5.2.3), while SalUn is the most competitive across all domains. This aligns with its widespread adoption in the literature, where it often serves as a reference baseline, and corroborates its reliability. In particular, GD and SRL emerge as the strongest methods overall (claim **c**): they outperform SCRUB (best DIS) on 7 out of 8 datasets and SSD (best WI) on all 8, a result we confirm statistically in Sect. 5.2.3. On the other hand, both DIS and WI methods show inconsistent performance. The former group performs best with smaller models (e.g., the tabular and graph domains). Still, the methods in DIS, especially BT, fail to scale to large pretrained architectures such as BERT, as distilling knowledge from such complex models is considerably more challenging, which is consistent with the literature (Marrie et al., 2024). Methods in WI, which directly modify model parameters, are particularly effective given the simpler architecture of MLPs compared to CNNs or LLMs, where kernels and attention mechanisms introduce unintended side effects.

Answer to RQ3. To what extent do existing unlearning methods generalize across domains, and are certain approaches consistently superior in specific application settings?
The main results of our benchmark can be summarized as follows:
(i). Simple FT methods are still the most reliable across the board. This means that methods like plain fine-tuning or negative gradient updates can still outperform more recent unlearning methods across the board. Unlearning remains fundamentally domain-dependent, as no method succeeds across all domains, although Fine-tuning methods are the most reliable.
(ii). A utility-efficacy trade-off is unavoidable with current methods, as the field lacks methods that forget without leaking information. DIS and WI methods either degrade utility, causing catastrophic forgetting, or expose the model to MIAs.
(iii). Building on **(ii)**, LUMA exposes hidden weaknesses: prior benchmarks overstated progress by ignoring efficiency and efficacy simultaneously.
(iv). Text classification is where most MU methods remain insufficiently effective, and where unlearning is particularly meaningful due to large model sizes.

5.2.3 Friedman-Wilcoxon Tests

To support our qualitative observations with statistical evidence, we apply the framework of Demšar 2006 for comparing multiple classifiers over multiple datasets. We use the mean LUMA score per (method, dataset) as the atomic observation (the tables shown in Sect. 5.2), yielding a 12×8 matrix. Using individual seed runs would constitute pseudoreplication, as seeds within the same (method, dataset) cell are replicates, not independent observations. Code for these tests is available in the repository.

We first apply the Friedman test, a non-parametric test that ranks methods within each dataset and tests whether the observed differences in average ranks across methods are larger than expected under random ranking. The test statistic measures how far the observed average ranks χ_F^2 deviate from this expectation.

Our null hypothesis H_0 is that all 12 methods are equivalent, i.e., any rank difference across datasets are due to randomness. A value near zero would indicate random ranking; a large value indicates systematic differences. We obtain $\chi_F^2 = 32.26$ ($p = 0.0007$), well above the critical value of 19.68 at $\alpha = 0.05$ (Any χ_F^2 above 19.68 would arise by chance less than 5% of the time under H_0; our value of 32.26 would arise less than 0.07% of the time). **This statistical result confirms that there are systematic differences among the 12 methods tested.**

For specific claims, we use the Wilcoxon signed-rank test: differently from Nemenyi or other post-hoc tests for targeted comparisons, they depend only on the two methods being compared (Benavoli et al., 2016). Results are reported in Table 6. Given two methods A and B, the test computes the signed differences $d_i = \mathrm{LUMA}_A^{(i)} - \mathrm{LUMA}_B^{(i)}$ across the N datasets, ranks them by absolute value, and computes: $W = \sum_{i:\, d_i > 0} \mathrm{rank}(|d_i|)$.

A large W indicates that A consistently outperforms B by larger margins than the reverse. We prove the three claims of Sect. 5.2.2: **(a.)** Within the FT family, GD and SRL significantly outperform ANG ($p < 0.05$). NG and UNSIR do not reach significance against ANG, reflecting the high variance of these methods across domains: NG achieves LUMA of 0.207 on CelebA and 0.138 on IMDB. Due to the number of datasets, non-significant results should be interpreted as inconclusive (i.e., we cannot reject the null hypothesis of equivalence). Still, ANG remains among the worst performers of the group. **(b.)** Within the SWM family, **CF-k and SalUn both significantly outperform EU-k** ($p < 0.01$), confirming that EU-k is consistently the weakest method in this group across all tested domains. **(c.)** GD and SRL outperform the strongest representatives of every other family: they beat SSD (best WI) on all 8 datasets and SCRUB (best DIS) on 7 out of 8, with all four comparisons surviving Holm correction. This provides statistical support for the main conclusion that FT methods are the most reliable across the benchmark (with the exception of ANG).

6 Conclusion

In this work, we bring order to the landscape of MU methods for classification by: **(i).** introducing LUMA, a unified metric that surpasses the existing ones in the literature, thus providing an easy way to compare unlearning methods and hyperparameters; **(ii).** building the most comprehensive benchmark to date, covering 4 data modalities, 12 unlearning

Table 6 Pairwise Wilcoxon signed-rank test results (Holm-Bonferroni corrected), one-sided tests (H_1: row > column)

Claim	Comparison	W	p_{raw}	p_{Holm}	Sig.
(a) ANG worst in FT	GD > ANG	35.0	.0078	.0234	*
	SRL > ANG	36.0	.0039	.0156	*
	NG > ANG	22.0	.3203	.3203	
	UNSIR > ANG	31.0	.0391	.0781	
(b) EU-*k* worst in SWM	CF-*k* > EU-*k*	36.0	.0039	.0078	*
	SalUn > EU-*k*	34.0	.0117	.0117	*
(c) GD, SRL best overall	GD > SSD	36.0	.0039	.0156	*
	SRL > SSD	36.0	.0039	.0156	*
	GD > SCRUB	33.0	.0195	.0195	*
	SRL > SCRUB	35.0	.0078	.0156	*

*Significant at $\alpha = 0.05$ after correction. $N = 8$ datasets

methods, 8 datasets, and 8 models; this includes the first systematic evaluation on the tabular modality. From this benchmark, we extract actionable guidelines for future evaluation of MU methods; moreover, we show results on best-performing unlearning methods across different modalities; and **(iii).** a publicly available, fully reproducible, and extensible benchmark to facilitate fair comparison in future work. Our results lead to clear guidelines for the design and deployment of unlearning techniques, and expose critical shortcomings that can only be revealed through cross-modality analysis. Together, these contributions establish a common ground for evaluating and comparing MU approaches for classification.

7 LLM Usage Disclosure

During the preparation of this work, the authors used LLMs to correct typos and grammatical mistakes. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Appendix A: LUMA Against Other Metrics

In this section, we show shortcomings and issues of the previous two unified unlearning metrics, GUM (Koudounas et al., 2025) and ToW (Zhao et al., 2024). Then, we detail how LUMA handles edge cases and its sensitivity to hyperparameters.

Shortcomings of Other Metrics

Indicating the Gold Model with g and the resulting Unlearned model with u, GUM is defined as:

$$\text{GUM} = \frac{(1+\alpha+\beta)UET}{\alpha ET + \beta UT + UE}$$

where $U = 1 - \left|F1_T^{(g)} - F1_T^{(u)}\right|$, $E = 1 - \left(\frac{\text{MIA}'(u) - \text{MIA}'(g)}{\text{MIA}(o) - \text{MIA}'(g)}\right)^2$, with

$$\text{MIA}'(u) = \min\{\text{MIA}(u), \text{MIA}(o)\}, \quad \text{MIA}'(g) = \min\left\{\text{MIA}(g), \frac{\text{MIA}'(u) + \text{MIA}(o)}{2}\right\}.$$

and lastly $T = 1 - \frac{\log\left(T^{(u)}+1\right)}{\log\left(T^{(g)}+1\right)}$.

While GUM includes all the three key evaluation dimensions (*ED*s) for MU methods (efficacy, utility, efficiency) it suffers from two core issues: *(i).* Only using one measure per dimension: the F1 score on the test set is not enough to assess the model's utility, and the RunTime is not enough to assess the MU method's efficiency. The F1 score on the forget set and the memory efficiency are also important. *(ii).* When $MIA'(u) - MIA(o) \leq \epsilon$ with a reasonably small ϵ, E grows without bound, effectively dominating GUM to unusable values. Worse, the metric is not computable at all for a division by 0 given two conditions:

- $MIA(g) \geq MIA(o)$
- $MIA'(u) = MIA(o)$, which happens if $MIA(u) \geq MIA(o)$

Consider the definition of E, specifically the denominator $MIA(o) - MIA'(g)$. This will be 0 when $MIA(o) = MIA'(g)$. Given the definition of $MIA'(g)$, this can only happen when $MIA(g) > MIA(o)$ (first condition) and $MIA'(u) = MIA(o)$. The latter is true when $MIA(u) = MIA(o)$ (second condition). We show an example in Section A.2.

Tug of War (ToW) is instead defined as:

$$\text{ToW}(\theta_u, \theta_g, S, R, D_{\text{test}}) = \big(1 - d_a(\theta_u, \theta_g, S)\big)\,\big(1 - d_a(\theta_u, \theta_g, R)\big)\,\big(1 - d_a(\theta_u, \theta_g, D_{\text{test}})\big),$$

where

$$a(\theta, D) = \frac{1}{|D|} \sum_{(x,y)\in D} 1[f(x;\theta) = y]$$

is the accuracy of a model f parameterized by θ on dataset D, and

$$d_a(\theta_u, \theta_r, D) = \big|\, a(\theta_u, D) - a(\theta_r, D) \,\big|$$

is the absolute difference between the accuracies of models θ_u (the unlearned model) and θ_g (the Gold Model) on the dataset D.

ToW includes multiple metrics for the Utility dimension, but misses the Efficiency dimension entirely. Accordingly, the Gold Model will always obtain the optimal ToW score of 1 (regardless of how expensive it is to train), and two unlearners that output the same retrained model with largely varying RunTimes will obtain the same ToW score. Moreover, by omitting MIA, ToW doesn't capture the information leakage of any MU method, which is a known problem in the literature (Le Quy et al., 2022; Xu et al., 2024).

Empirical Evidence

Table 7 is an extract from the bigger Table 3 shown in Sect. 5. Reporting UMIA instead of MIA is functionally the same for this proof. This is empirical evidence of the proven problem of GUM in Section A.1: we have that $MIA(g) \geq MIA(o)$ as $0.499 \geq 0.498$ (first condition) and that $MIA'(u) = MIA(o)$ as $0.497 = 0.497$ (see definition of $MIA'(u)$ above). For this reason, GUM cannot be calculated and will fail due to statistical variation in the *MIA* scores.

GUM is very useful because it combines measures across all *ED*s, but suffers from numerical instability under some conditions. LUMA fixes this by employing Laplacian kernels. ToW, on the other hand, suffers from the opposite problem: ignoring Efficiency

Table 7 Extract of results on the Adult dataset (Table 3)

Group	Method	Adult			
		F1 test	UMIA	LUMA	GUM
–	`Orig.`	$.790 \pm .001$	$.497 \pm .002$	$.633 \pm .002$	X
–	`Gold`	$.790 \pm .002$	$.499 \pm .002$	$.636 \pm .000$	X
`FT`	`GD`	$.790 \pm .001$	$.498 \pm .001$	$.693 \pm .002$	X

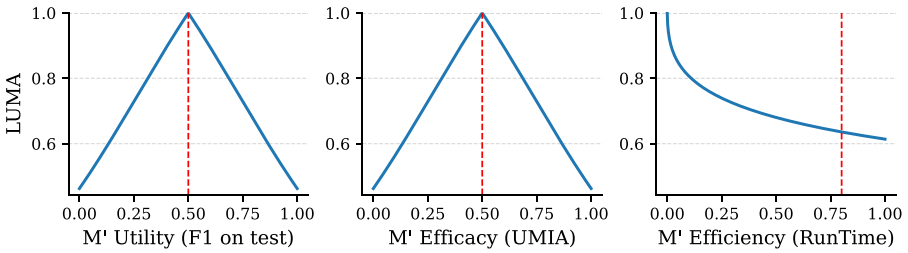

Fig. 3 Sensitivity of the proposed LUMA metric with respect to each *ED*

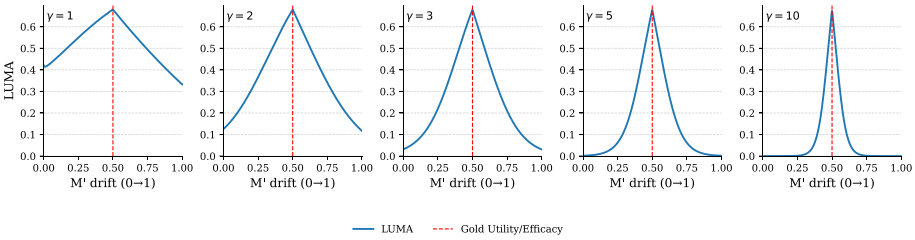

Fig. 4 Effect of the Laplacian kernel parameter γ on the LUMA metric as a model drifts from an all-zero representation to an all-one representation of the *ED*s of Utility and Efficacy

entirely. Consider a toy Unlearner that simply retrains the model from scratch but deliberately takes twice as long. Despite the wasted resources, it would still achieve a perfect ToW score. This illustrates that, to serve as a truly unified metric for unlearning, the *ED* dimension of Efficiency must be incorporated.

LUMA 's Hyperparameters

LUMA fixes all shortcomings of other metrics shown in Sect. 4 by considering of all the three dimensions, possibly with more than one metric each. By employing Laplacian kernels, LUMA is easily extendable with future MU measures (Fig. 3).

By construction, the Laplacian kernels ensure that both positive and negative drifts in an evaluation dimension are penalized symmetrically. Figure 4 illustrates this property, showing the response of LUMA with respect to each *ED* while assuming perfect performance on the remaining *ED*s. For Utility and Efficacy, LUMA reaches its maximum value of 1 when the varying *ED* matches the performance of `Gold`, and decreases smoothly as the model drifts away in either direction. For Efficiency, LUMA decreases inversely with runtime, lowering as runtime increases.

The Laplacian kernels are parametrized by γ. We chose $\gamma = 3$ to make LUMA sensitive to even small single-measure drifts between the Gold Model (`Gold`) and the unlearned model (M'). Empirically, $\gamma = 3$ provides a useful middle ground: lower values underpenalize deviations, while larger values of γ lead to overly sharp behavior, with scores failing to exploit the full range of the metric.

Figure 4 illustrates the impact of the kernel parameter γ on LUMA as M' drifts from 0 to 1 across all measures. For reference, the Gold model is fixed at 0.5 on all measures. In this work, we chose $\gamma = 3$ as it provided the best smoothness.

Moreover, LUMA is customizable in terms of weights assigned to measures in the Efficiency *ED*. The vast majority of works in the Machine Unlearning literature only consider RunTime when evaluating Efficiency, discarding peak memory usage. In this work, we introduced memory usage as a measure with weight = 0.5. This weight vector can be customized according to available resources and the task at hand. Figure 5 shows the sensitivy of LUMA to the weight vector w when the runtime scales w.r.t. to the Gold Model, assuming equality on the Utility and Efficacy *ED*s. The figure corroborates the correctness of LUMA: if we assign weight 1 to RunTime, and the RunTime is 0, LUMA will be 1, although this is basically unachievable in practice. The vector w, weighting Efficiency measures, is customizable. In this study we set $w = [0.5, 0.5]$. Figure 5 shows that this choice gives runtime decisive influence over the score while still ensuring memory is not ignored.

Summing up, *(i)*. By employing Laplacian kernels, LUMA fixes the shortcomings of GUM for edge cases, and surpasses both GUM and ToW in considering more than one measure per *ED*and being easily extensible with other metrics. *(ii.)* By incorporating all *ED*s, LUMA surpasses ToW in completeness, as the latter did not consider Efficiency at all.

Appendix B: Benchmark Details

In this section, we show complete details on the Datasets, forget sets, Models, and Unlearners we employed in our benchmark. Moreover, in Section B.3, we show how to easily extend the benchmark for future work.

All details and a step-by-step guide can be found at this repository.

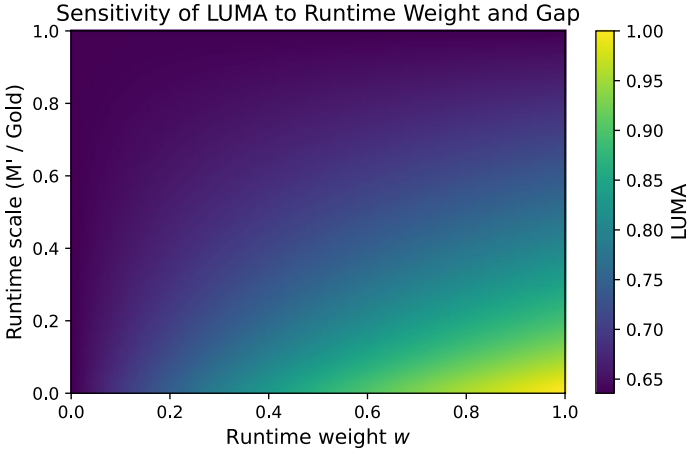

Fig. 5 Heatmap of LUMA values as a function of runtime weight w (x-axis) and runtime scale of M' relative to the Gold model (y-axis). The weight assigned to memory is $1 - w$

Datasets

Table 8 shows the number of samples, the number of attributes, the number of classes, and the domain for each dataset. In the Image domain, we employed CIFAR-100 for multiclass classification and CelebA for binary multilabel classification. As detailed in Sect. 3, we selected these datasets as the most prominent in the Unlearning literature for each domain.

Models

Table 9 shows the general architecture of the model we employed. Our rationale was to choose two models per domain with a similar architecture but different sizes. Full configurations can be found at this repository.

Extending the Framework

Our benchmark was built to be easily extendable by researchers and practitioners. New experiments can be defined by just extending the configuration files. A full, step-by-step demonstration on how to implement a new Unlearner method is available at this repository.

Appendix C: Additional Results

In this Section, we report the additional results of the sensitivity analysis shown in the main paper with the complete results across all experiments.

Table 8 Overview of the datasets employed for our benchmark

Dataset	Domain	Features	Samples	Classes
Adult	Tabular	14 tabular features	48,842	2
Spotify	Tabular	15 tabular features	114,000	8
Cifar100	Image	32x32 color images	60,000	100
CelebA	Image	178×218 color images	202,599	40 (ml*)
AG news	Text	Textual descriptions	127,600	4
IMDB	Text	Textual descriptions	100,000	2
BACE	Graph	Graphs (∼34N, ∼74E, 9F)	1513	2
BBBP	Graph	Graphs (∼23N, ∼51E, 9F)	2,050	2

*ml stands for Multilabel. For Graphs, N stands for nodes, E for edges and F for features

Table 9 Overview of the models employed for our benchmark

Architecture	Domain	Epochs	Optimizer	LR
1-hidden layer MLP	Tabular	2	Adam	1e-3
3-hidden layer MLP	Tabular	2	Adam	1e-3
ResNet-18	Image	5	Adam	1e-3
ResNet-50	Image	5	Adam	1e-3
DistilBERT	Text	3	AdamW	2e-5
BERT	Text	3	AdamW	2e-5
1-layer GCN	Graph	50	RMSprop	1e-3
2-layer GCN	Graph	50	RMSprop	1e-3

Sensitivity Analysis Additional Results

The following tables report a systematic sensitivity analysis of the unlearning performance across key experimental factors. For each dataset, results are presented in terms of LUMA, Membership Inference Attack score, test F1 score, and runtime. The analyses examine how these metrics vary with respect to three main dimensions: the learning rate of the unlearning methods, the model dimension, and the size of the forget set. Each metric is reported in a dedicated column, while different unlearning strategies are grouped by method family and compared against the original model and the corresponding Gold retrained model (Figs. 6, 7, 8, 9, 10).

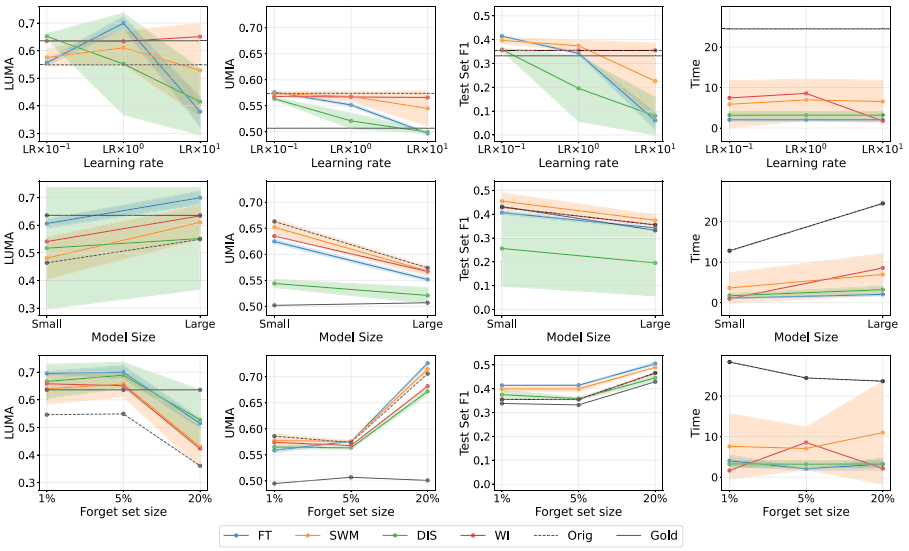

Fig. 6 Sensitivity analysis of unlearning performance across key experimental factors on the Cifar100 dataset

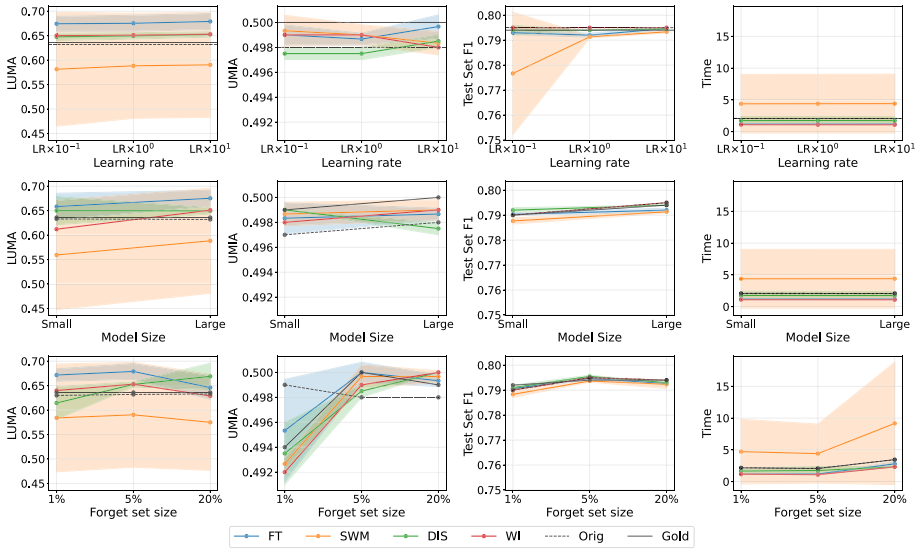

Fig. 7 Sensitivity analysis of unlearning performance across key experimental factors on the Adult dataset

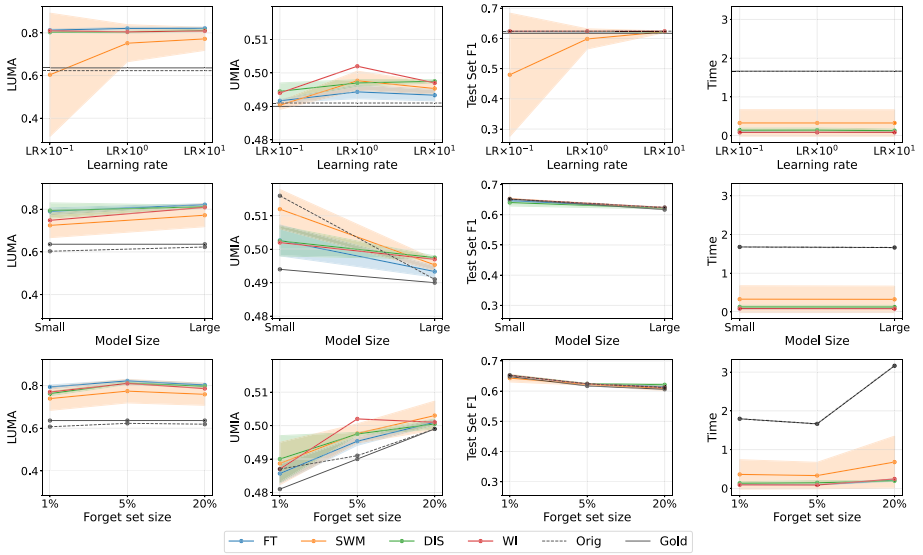

Fig. 8 Sensitivity analysis of unlearning performance across key experimental factors on the Spotify dataset

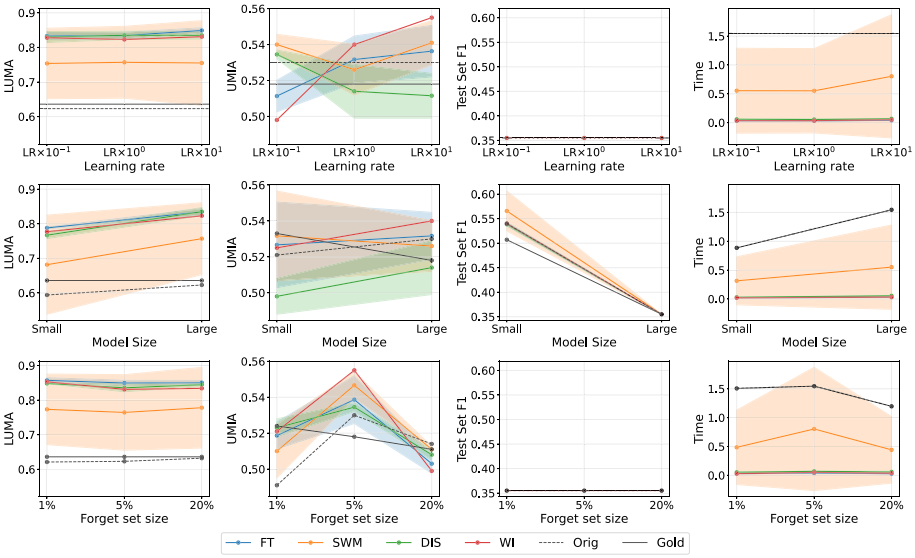

Fig. 9 Sensitivity analysis of unlearning performance across key experimental factors on the BACE dataset

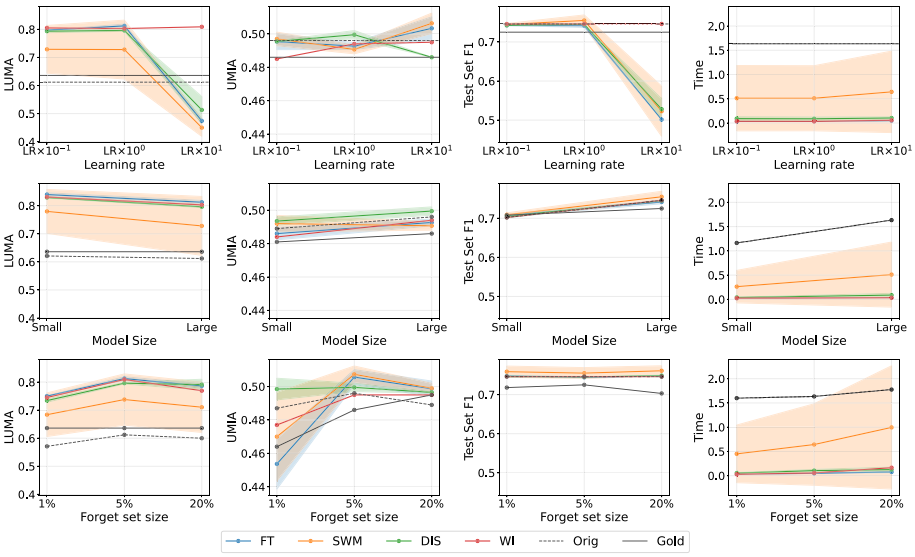

Fig. 10 Sensitivity analysis of unlearning performance across key experimental factors on the BBBP dataset

Luma Results

See Tables 10, 11.

Table 10 LUMA scores for Large (L) and Small (S) models with two forget set sizes

Group	Alg.	Cifar100				CelebA			
		S-5	L-5	S-20	L-20	S-5	L-5	S-20	L-20
	Orig.	.464	.549	.395	.361	.501	.460	.496	.502
	Gold	.636	.636	.636	.636	.636	.636	.636	.636
FT	GD	.617	.715	.561	.525	.666	.709	.674	.654
	SRL	.619	.719	.555	.531	.730	.676	.671	.633
	NG	.553	.522	.200	.192	.240	.207	.196	.225
	ANG	.216	.289	.374	.375	.109	.089	.079	.097
	UNSIR	.583	.665	.510	.488	.624	.613	.650	.560
SWM	CF-k	.522	.698	.430	.421	.478	.495	.607	.608
	EU-k	.440	.592	.370	.340	.428	.456	.546	.553
	SalUn	.585	.682	.479	.530	.680	.608	.617	.641
DIS	BT	.737	.736	.448	.636	.652	.699	.604	.672
	SCRUB	.550	.642	.452	.419	.608	.542	.644	.618
WI	FF	.360	.307	.469	.401	.774	.759	.749	.666
	SSD	.541	.651	.434	.423	.568	.561	.576	.585
		AG News				**IMDB**			
	Orig.	.561	.472	.575	.578	.587	.588	.588	.593
	Gold	.636	.636	.636	.636	.636	.636	.636	.636
FT	GD	.606	.581	.692	.650	.662	.665	.659	.643
	SRL	.634	.593	.670	.617	.626	.649	.630	.625
	NG	.505	.289	.037	.031	.332	.138	.077	.074
	ANG	.260	.339	.623	.244	.602	.643	.607	.072
	UNSIR	.615	.593	.645	.666	.680	.639	.650	.606
SWM	CF-k	.642	.597	.669	.669	.681	.693	.678	.685
	EU-k	.564	.559	.621	.621	.647	.666	.659	.627
	SalUn	.608	.578	.643	.602	.650	.627	.646	.615
DIS	BT	.160	.181	.440	.211	.531	.614	.656	.645
	SCRUB	.583	.606	.639	.618	.659	.630	.620	.641
WI	FF	.039	.044	.039	.035	.124	.184	.110	.146
	SSD	.589	.514	.602	.615	.636	.637	.681	.631

Table 11 LUMA scores for Large (L) and Small (S) models with two forget set sizes

Group	Alg.	Adult				Spotify			
		S-5	L-5	S-20	L-20	S-5	L-5	S-20	L-20
	Orig.	.632	.633	.633	.633	.623	.603	.619	.592
	Gold	.636	.636	.636	.636	.636	.636	.636	.636
FT	GD	.696	.693	.676	.673	.827	.804	.812	.787
	SRL	.684	.672	.646	.633	.824	.799	.804	.773
	NG	.837	.823	.706	.262	.836	.799	.825	.787
	ANG	.592	.554	.573	.546	.794	.761	.781	.765
	UNSIR	.657	.623	.616	.593	.815	.768	.792	.739
SWM	CF-k	.683	.669	.665	.649	.818	.777	.805	.756
	EU-k	.440	.408	.438	.398	.696	.657	.687	.642
	SalUn	.648	.607	.622	.611	.808	.752	.785	.721
DIS	BT	.647	.689	.696	.696	.805	.830	.804	.811
	SCRUB	.658	.621	.642	.605	.818	.758	.790	.742
WI	FF	.812	.776	.860	.870	.785	.708	.802	.650
	SSD	.653	.613	.629	.632	.810	.748	.786	.717
		BACE				**BBBP**			
	Orig.	.594	.623	.586	.632	.621	.612	.623	.600
	Gold	.636	.636	.636	.636	.636	.636	.636	.636
FT	GD	.785	.847	.787	.859	.842	.816	.827	.797
	SRL	.791	.860	.789	.849	.846	.818	.824	.778
	NG	.647	.854	.601	.852	.751	.769	.498	.401
	ANG	.684	.852	.826	.849	.828	.797	.823	.806
	UNSIR	.793	.841	.782	.843	.835	.803	.821	.783
SWM	CF-k	.791	.847	.779	.856	.839	.811	.828	.791
	EU-k	.558	.610	.603	.613	.674	.609	.682	.585
	SalUn	.789	.836	.782	.865	.832	.793	.811	.755
DIS	BT	.786	.847	.540	.850	.834	.799	.837	.810
	SCRUB	.775	.824	.770	.839	.832	.793	.813	.771
WI	FF	.734	.788	.738	.787	.732	.758	.717	.707
	SSD	.777	.831	.761	.834	.831	.809	.822	.769

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10994-026-07094-y.

Author Contributions A.D. and C.S. contributed equally to this work. They ran the experiments, wrote the first draft, wrote the code, built the visualizations, reviewed the manuscript. F.G. wrote the manuscript text and reviewed it. E.B. reviewed the manuscript. G.S. helped writing the code and reviewing the manuscript.

Funding Open access funding provided by Università degli Studi dell'Aquila within the CRUI-CARE Agreement.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Becker, B., & Kohavi, R. (1996). Adult. UCI Machine Learning Repository. https://doi.org/10.24432/C5XW20

Benavoli, A., Corani, G., & Mangili, F. (2016). Should we really use post-hoc tests based on mean-ranks? *Journal of Machine Learning Research, 17*(1), 152–161.

Blanco-Justicia, A., Jebreel, N., Manzanares-Salor, B., Sánchez, D., Domingo-Ferrer, J., Collell, G., & Eeik Tan, K. (2025). Digital forgetting in large language models: A survey of unlearning methods. *Artificial Intelligence Review, 58*(3), 90.

Cadet, X. F., Borovykh, A., Malekzadeh, M., Ahmadi-Abhari, S., & Haddadi, H. (2024). Deep unlearn: Benchmarking machine unlearning arXiv:2410.01276 arXiv preprint.

Cha, S., Cho, S., Hwang, D., Lee, H., Moon, T., & Lee, M. (2024). Learning to unlearn: instance-wise unlearning for pre-trained classifiers. *AAAI Press*. https://doi.org/10.1609/aaai.v38i10.28996

Cheng, J., & Amiri, H. (2024). Mu-bench: A multitask multimodal benchmark for machine unlearning arXiv:2406.14796 arXiv preprint.

Choi, D., & Na, D. (2023). Towards machine unlearning benchmarks: Forgetting the personal identities in facial recognition systems arXiv:2311.02240 arXiv preprint.

Chundawat, V. S., Tarun, A. K., Mandal, M., & Kankanhalli, M. (2023). Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. Proceedings of the AAAI Conference on Artificial Intelligence

Crawford, K. (2022). Atlas of AI: Power, politics, and the planetary costs of artificial intelligence. Yale University Press.

D'Angelo, A., Savelli, C., Tagliente, G., Giobergia, F., Baralis, E., & Stilo, G. (2025a). ERASURE: A modular and extensible framework for machine unlearning. ACM

D'Angelo, A., Savelli, C., Tagliente, G., Giobergia, F., Baralis, E., & Stilo, G. (2025b). How to make reproducible research in machine unlearning with ERASURE. In: International joint conferences on artificial intelligence organization. Demo Track . https://doi.org/10.24963/ijcai.2025/1255

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*(1), 1–30.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding arXiv:1810.04805.

Fan, C., Liu, J., Hero, A., & Liu, S. (2024). Challenging forgets: Unveiling the worst-case forget sets in machine unlearning. In: European conference on computer vision, pp. 278–297. Springer

Fan, C., Liu, J., Zhang, Y., Wong, E., Wei, D., & Liu, S. (2023). Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. arXiv preprint arXiv:2310.12508

Foster, J., Schoepf, S., & Brintrup, A. (2024). Fast machine unlearning without retraining through selective synaptic dampening. *Proceedings of the AAAI conference on artificial intelligence* (Vol. 38, pp. 12043–12051)

Giobergia, F. (2023). IMDb-ID Dataset. https://huggingface.co/datasets/fgiobergia/imdb-id. Accessed: 2026-01-10

Goel, S., Prabhu, A., Sanyal, A., Lim, S.-N., Torr, P., & Kumaraguru, P. (2022). Towards adversarial evaluations for inexact machine unlearning arXiv:2201.06640 arXiv preprint.

Golatkar, A., Achille, A., & Soatto, S. (2020). Eternal sunshine of the spotless net: Selective forgetting in deep networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9304–9312)

Grimes, K., Abidi, C., Frank, C., & Gallagher, S. (2024). Gone but not forgotten: Improved benchmarks for machine unlearning arXiv:2405.19211 arXiv preprint.
Grynbaum, M. M., & Mac, R. (2023). The times sues openai and microsoft over ai use of copyrighted work. The New York Times, 27.
Gulli, A. (2005). *AG's Corpus of news articles* (pp. 2026–2027)
Hayes, J., Shumailov, I., Triantafillou, E., Khalifa, A., & Papernot, N. (2024). Inexact unlearning needs more careful evaluations to avoid a false sense of privacy arXiv:2403.01218 arXiv preprint.
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778)
Jung, D. (2025). Entun: Mitigating the forget-retain dilemma in unlearning via entropy. ICT Express.
Koudounas, A., Savelli, C., Giobergia, F., & Baralis, E. (2025). Alexa, can you forget me. *Machine unlearning benchmark in spoken language understanding* (pp. 2025–2607). https://doi.org/10.21437/Interspeech Interspeech 2025.
Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. University of Toronto. Technical report.
Kurmanji, M., Triantafillou, P., Hayes, J., & Triantafillou, E. (2024). Towards unbounded machine unlearning. Advances in Neural Information Processing Systems **36**
Lanyon, J., Finke, A., Andreou, P., & Cosma, G. (2025). On the limitation of evaluating machine unlearning using only a single training seed arXiv:abs/2510.26714.
Le Quy, T., Roy, A., Iosifidis, V., Zhang, W., & Ntoutsi, E. (2022). A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 12*(3), 1452.
Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. Proceedings of international conference on computer vision (ICCV)
Maharshipandya: Spotify tracks dataset. https://huggingface.co/datasets/maharshipandya/spotify-tracks-dataset (2023)
Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., & Kolter, J. Z. (2024). Tofu: A task of fictitious unlearning for llms arXiv:2401.06121 arXiv preprint.
Mantelero, A. (2013). 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.
Marrie, J., Arbel, M., Mairal, J., & Larlus, D. (2024). On good practices for task-specific distillation of large pretrained visual models
Sakiyama, H., Fukuda, M., & Okuno, Y. (2021). Prediction of blood-brain barrier penetration (bbbp) based on molecular descriptors of the free-form and in-blood-form datasets. *Molecules, 26*(24), 7428. https://doi.org/10.3390/molecules26247428
Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter arXiv:1910.01108.
Savelli, C., La Quatra, M., Koudounas, A., & Giobergia, F. (2026). FAME: Fictional actors for multilingual erasure. Proceedings of the fifteenth language resources and evaluation conference european language resources association.
Tarun, A. K., Chundawat, V. S., Mandal, M., & Kankanhalli, M. (2023). Fast yet effective machine unlearning. IEEE transactions on neural networks and learning systems
Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., & Pande, V. (2018). Moleculenet: a benchmark for molecular machine learning. *Chemical Science, 9*(2), 513–530. https://doi.org/10.1039/C7SC02664A
Xu, H., Zhu, T., Zhang, L., Zhou, W., & Yu, P. S. (2024). Machine unlearning: A survey
Yu, L., Zhao, Z., Wang, Y., Wang, P., Cao, X., Wang, B., & Wang, Y. (2026). Falw: A forgetting-aware loss reweighting for long-tailed unlearning. arXiv preprint arXiv:2601.18650
Zhao, K., Kurmanji, M., Barbulescu, G.-O., Triantafillou, E., & Triantafillou, P. (2024). What makes unlearning hard and what to do about it. In: Advances in neural information processing systems (NeurIPS 2024)

Authors and Affiliations

Andrea D'Angelo[1,4] · Claudio Savelli[2] · Flavio Giobergia[2] · Elena Baralis[2] · Giovanni Stilo[3]

✉ Andrea D'Angelo
andrea.dangelo6@graduate.univaq.it

Claudio Savelli
claudio.savelli@polito.it

Flavio Giobergia
flavio.giobergia@polito.it

Elena Baralis
elena.baralis@polito.it

Giovanni Stilo
gstilo@luiss.it

[1] University of L'Aquila, L'Aquila, Italy

[2] Polytechnic University of Turin, Turin, Italy

[3] Luiss Guido Carli University of Rome, Rome, Italy

[4] Present address: Aarhus University, Aarhus N, Denmark