OpenUnlearning: Accelerating LLM Unlearning via Unified Benchmarking of Methods and Metrics

Vineeth Dorna [*1] Anmol Mekala [*1] Wenlong Zhao [1] Andrew McCallum [1]
Zachary C. Lipton [2] J. Zico Kolter [2] Pratyush Maini [2]

Abstract

Deploying large language models (LLMs) raises concerns about privacy, safety, and legal compliance due to their tendency to memorize sensitive content. Robust unlearning is essential to ensure their safe and responsible use. Yet the task is inherently challenging, partly due to difficulties in reliably measuring whether unlearning has truly occurred. Moreover, fragmentation in current methodologies and inconsistent evaluation metrics hinder comparative analysis and reproducibility. To unify and accelerate research efforts, we introduce `OpenUnlearning`, a standardized and extensible framework that supports a wide range of unlearning methods, metrics, and benchmarks, enabling comprehensive evaluation. Leveraging `OpenUnlearning`, we propose a novel meta-evaluation benchmark focused specifically on assessing the faithfulness and robustness of evaluation metrics themselves. Overall, we establish a clear, community-driven pathway toward rigorous development in LLM unlearning research.

1. Introduction

LLMs often memorize sensitive, copyrighted or harmful content from their vast training data, raising privacy (Carlini et al., 2023), safety (Wei et al., 2023) and legal (Karamolegkou et al., 2023; Union, 2016; OAG, 2021) concerns. Ever increasing costs of pre-training and post-training (Grattafiori et al., 2024; Team et al., 2023; 2024) prevent re-training in response to deletion requests (Liu et al., 2024). This has motivated the development of machine *unlearning* techniques that allow for "forgetting" training data via efficient post-training interventions (Nguyen et al., 2022; Liu et al., 2024). The goal of unlearning is to eliminate the undesirable influences from specific training data, while maintaining the overall behavior and performance.

There has been a recent surge in LLM unlearning research, yielding numerous proposed methods on several benchmarks. Modifying model weights to achieve unlearning is of the most interest, with many proposed appproaches (Zhang et al., 2024; Wang et al., 2025c; Li et al., 2024; Fan et al., 2024; Mekala et al., 2025; Dong et al., 2025; Jia et al., 2024; Wang et al., 2025b; Fan et al., 2025). Concurrently, several benchmarks have been proposed to evaluate unlearning across a wide range of setups, covering aspects such as synthetic fine-grained unlearning, open-ended unlearning, knowledge, PII, memorization and privacy focused unlearning (Maini et al., 2024; Qiu et al., 2024; Ramakrishna et al., 2025a; Shi et al., 2025; Li et al., 2024; Qiu et al., 2024; Tian et al., 2024; Jin et al., 2024; Eldan & Russinovich, 2023). This volume of LLM unlearning research is marked by a notable fragmentation. Different benchmarks use different evaluations, with no consensus on the best evaluations and considerable criticism of existing evaluations (Thaker et al., 2025; Scholten et al., 2024; Wang et al., 2025a; Zhang et al., 2025b; Doshi & Stickland, 2024; Lynch et al., 2024). Evaluating unlearning is a nuanced task involving knowledge, privacy, and utility desiderata, which is arguably as hard as achieving unlearning itself (Schwarzschild et al., 2024; Łucki et al., 2025). To catalyze research efforts, we present

A unifying framework. We introduce `OpenUnlearning`, a standardized and extensible framework that unifies research efforts by integrating 3 widely used benchmarks, 9 unlearning algorithms, and 16 evaluation metrics under different stress testing scenarios. Through this standardized framework, we foster unified research efforts and expedite the creation of effective unlearning techniques and benchmarks.

Evaluating evaluations. Leveraging `OpenUnlearning`, we conduct the first meta-evaluation of unlearning metrics using 450+ open-sourced models with known ground truth states. We compare 12 forgetting metrics against desiderata for faithfulness and robustness, establishing a benchmark to guide future improvements in evaluation.

[*]Equal contribution [1]University of Massachusetts Amherst, Amherst, MA, USA [2]Carnegie Mellon University, Pittsburgh, PA, USA. Correspondence to: Vineeth Dorna <vdorna@umass.edu>, Anmol Mekala <amekala@umass.edu>.

Published at MemFM Workshop at ICML 2025, Vancouver, Canada.

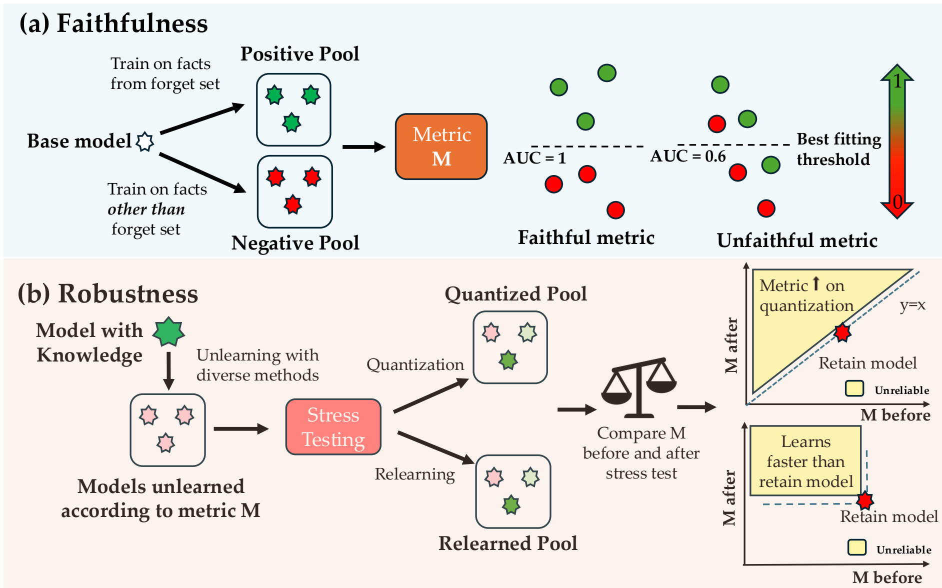

Figure 1. **Meta-evaluation of unlearning metrics**: (1) *Faithfulness*: the metric distinguishes models with and without target knowledge, reflected by high AUC; (2) *Robustness*: the metric value does not increase under benign changes (e.g., quantization) and does not improve faster than a retain model under non-benign changes (e.g., relearning).

2. OpenUnlearning

We envision LLM unlearning evolving within a shared framework that continuously integrates new and improved methods and evaluations—where unlearning methods iteratively improve on benchmarks, and evaluation metrics themselves improve through meta-evaluation and critical feedback. Despite growing research, LLM unlearning lacks unified technical and conceptual foundations. Fragmented benchmarks and inconsistent implementations hinder reproducibility, and progress. Moreover, unlearning methods and evaluation metrics aren't consistently extended across benchmarks, preventing standardization and comparative analysis. We illustrate this fragmentation across key parts of the unlearning pipeline:

1. **Benchmark fragmentation:** New methods are not implemented in all benchmarks. For example: UNDIAL (Dong et al., 2025) is not implemented on any of TOFU (Maini et al., 2024), MUSE (Shi et al., 2025) and WMDP (Li et al., 2024) . Similarly, evaluation metrics like MIA from MUSE are not implemented in TOFU; and LM Eval Harness benchmarks (Gao et al., 2024c) used in WMDP can be extended to TOFU, MUSE.
2. **Disparate method integrations:** Several approaches involve custom loss functions (Zhang et al., 2024; Maini et al., 2024; Fan et al., 2024; Dong et al., 2025) and others make adjustments to optimization steps (Wang et al., 2025b; Jia et al., 2024; Fan et al., 2025). These techniques could be modularized and reused across tasks for deeper investigation and a fair comparison.
3. **Duplicated core components:** Evaluation metrics often reuse core components (e.g., probabilities, ROUGE, MIA), yet implementations remain isolated. Preprocessing is duplicated across benchmarks despite shared formats (e.g., WMDP/MUSE corpora, TOFU/RWKU chat-style prompts). Stress tests for robustness, though proposed in some work, could also be standardized across benchmarks.

To address this, we introduce `OpenUnlearning`: a unified, extensible pipeline that consolidates benchmarks, methods, evaluation metrics, datasets, and stress-tests under one roof (see Figure 3) to accelerate unlearning research. More details can be found in Appendix D.

3. Evaluating Unlearning Evaluations

Reliable evaluations for unlearning are essential for regulatory compliance and data privacy, yet remain challenging (Kim et al., 2025; Schwarzschild et al., 2024; Łucki et al., 2025), especially for LLMs, due to ambiguity between memorization and generalization. We propose two minimal necessary desiderata—*Faithfulness* and *Robustness*—guided by our meta-evaluation framework, to promote trustworthy unlearning metrics (outlined in Figure 1).

Our meta-evaluation of metrics uses a test-bed of models with known ground truths. We employ the TOFU benchmark (Maini et al., 2024) with the improvements described from Appendix D.3 with the `forget10` unlearning task (forgetting 10% of TOFU). We use the LLAMA-3.2 1B model (Grattafiori et al., 2024), analyzing 12 unlearning

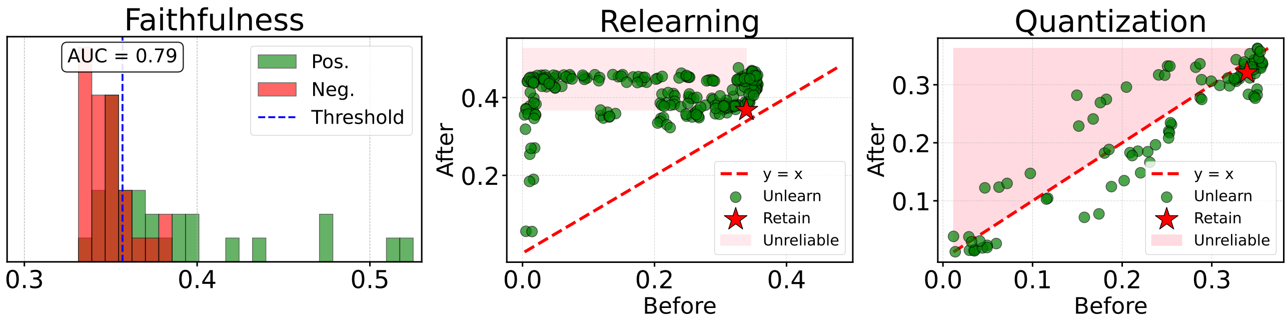

Figure 2. For the ROUGE metric we evaluate faithfulness (left) and robustness to quantization (middle), and relearning (right). Faithfulness achieves an AUC of 0.79, indicating substantial prediction overlap between models trained with and without the target knowledge. Relearning robustness is 0.48, showing many unlearned models re-acquire knowledge faster than the retain model upon re-exposure. Quantization robustness is 0.93, reflecting no distinctive trend of metric spikes post-quantization.

metrics adjusted to $[0, 1]$ scale (see Appendix D.3).

3.1. Faithfulness

Faithfulness

Motivation. Unlearning evaluations may not faithfully reflect an LLM's knowledge.
Desideratum. A faithful metric accurately reflects the presence of targeted knowledge by assigning consistently higher scores to models possessing it than to those lacking it.

LLMs often fail to regurgitate facts that remain encoded in their parameters when prompted, making it hard to tell whether a model truly forgot a target fact or simply refrained from exposing it (Doshi & Stickland, 2024; Lynch et al., 2024; Scholten et al., 2024; Wang et al., 2025a). For example, work by Doshi & Stickland (2024) shows that simple paraphrasing of inputs can yield a tenfold increase in evaluation scores on 'unlearned' models, indicating that the apparent forgetting may only be superficial. "Deeper" evaluation metrics aim to quantify this knowledge more faithfully, like Truth Ratio (Maini et al., 2024), GCG (Gandikota et al., 2024), or by using prompt engineering (Wang et al., 2025a; Shostack, 2024; Thaker et al., 2025).

On the other hand, evaluation metrics can register misleadingly high scores without the presence of the target knowledge (Maini et al., 2024). For example, in a question-answering evaluation using metrics like ROUGE, a model might achieve a high score by closely matching an expected answer template, even without correctly recalling the specific target fact. This calls for metrics that are **faithful** to the knowledge encoded in the model weights.

We measure *faithfulness* as the ability of metrics to distinguish models trained with (*positive pool*, P) versus without (*negative pool*, N) target knowledge: (i) Each pool has 30 diverse models trained under varying conditions. (ii) These variants present the target `forget10` information for pool P models in diverse, challenging formats (e.g., biography vs. QA, paraphrases). Pool N models serve as negative controls, using similarly structured data lacking this target information using various perturbations and alternative datasets. (iii) Metric scores yield two distributions (for P and N), and we compute AUC-ROC to quantify their separability. (iv) We select a classification threshold optimizing accuracy, which is subsequently used in robustness tests.

3.2. Robustness

Robustness

Motivation. Unlearning evaluations can be vulnerable to stress-testing interventions.
Desideratum. A robust metric's positive assessment of unlearning should (1) not flip upon benign model interventions; and (2) behave comparably to a model truly unfamiliar with the data under non-benign interventions.

Robustness of unlearning metrics is probed using various stress-test interventions. These include: (1) relearning attempts, where the unlearned model is further trained to potentially recover the forgotten information (Lynch et al., 2024; Hu et al., 2025; Łucki et al., 2025; Wang et al., 2025a); and (2) applying techniques like quantization (Zhang et al., 2025b). These have revealed the unreliability of several unlearning evaluation metrics.

For example, Zhang et al. (2025b) show that the PrivLeak metric (Shi et al., 2025) that previously reported a model as successfully unlearned can effectively 'flip' after a benign intervention, revealing that the targeted knowledge was perhaps never truly erased (Zhang et al., 2025b). Such significant fluctuations under stress tests undermine the reliability of evaluation metrics. Furthermore, models unlearned with respect to a metric can exhibit high susceptibility on metric evaluation to non-benign interventions like relearning, where evaluation metrics show an unusually rapid return of the supposedly forgotten knowledge even with minimal retraining effort (Fan et al., 2025; 2024). Robustness assesses stability under interventions like relearning, and quantization.

Robustness to Relearning: We evaluate metric scores before (m^a) and after (m^b) relearning on forget-set data. Then, we compare relative metric score recovery rates between unlearned (m_{unl}) and retain (m_{ret}) models, where higher R implies greater robustness.

$$r = \frac{m^a_{\text{ret}} - m^b_{\text{ret}}}{m^a_{\text{unl}} - m^b_{\text{unl}}}, \quad R = \min(r, 1). \tag{1}$$

Robustness to Quantization: We quantize models to 4-bit precision and compute scores before and after quantization, where higher Q implies greater robustness.

$$q = \frac{m^b_{\text{unl}}}{m^a_{\text{unl}}}, \quad Q = \min(q, 1). \tag{2}$$

3.3. Realistic Model Filtering

We enforce practical constraints by filtering models with: (i) Utility drops exceeding 20%. (ii) Insufficient unlearning w.r.t. the considered metric (more than the threshold computed in §3.1's *faithfulness* analysis). Models with substantial utility drop are unusable, while those without evidence of unlearning offer little insight into robustness. Our analysis focuses on ∼400 diverse models spanning eight unlearning methods — GradDiff, IdkDPO, IdkNLL (Maini et al., 2024), NPO (Zhang et al., 2024), SimNPO (Fan et al., 2024), AltPO (Mekala et al., 2025), UNDIAL (Dong et al., 2025) and RMU (Li et al., 2024) (more details in Appendix D.7, G.3).

3.4. Aggregation of Metrics

A effective unlearning metric must be both faithful and robust—e.g., a metric that consistently labels models as unlearned may be robust but not faithful. To penalize imbalance and reward well-rounded performance, we aggregate scores using the harmonic mean (HM). Figure 2 shows this for ROUGE; see Appendix F.4 for details and comparisons.

$$\text{Robustness} = \text{HM}(R, Q) \tag{3}$$

$$\text{Overall} = \text{HM}(\text{Faithfulness}, \text{Robustness}) \tag{4}$$

3.5. Results and Discussion

Table 1 highlights key insights: (i) **Extraction Strength (ES)** (Carlini et al., 2021) emerges as most reliable overall, aligning with Wang et al. (2025a). (ii) **Truth Ratio** has superior faithfulness but lower robustness, ranking third overall. (iii) Metrics based on raw probabilities or ROUGE scores have moderate faithfulness and robustness, limiting their reliability. (iv) Membership inference (MIA)-based metrics demonstrate high faithfulness but lack robustness, cautioning against relying solely on MIA metrics for assessing unlearning. This sensitivity raises concerns about the reliability of the MIA-based privacy assessments in unlearning contexts as introduced by Shi et al. (2025), as even benign interventions can reverse unlearning effects, as observed in Zhang et al. (2025b).

Table 1. Meta-evaluation of 12 unlearning metrics for Faithfulness and Robustness. Robustness is assessed using two stress-testing methods: quantization and relearning, with their harmonic mean reported as Agg. An overall aggregation across both Faithfulness and Robustness is reported in the first Agg. column. Higher scores indicate better performance (↑) in all dimensions. The best values are shown in bold, and the second-best values are underlined.

Metrics	Agg. ↑	Faithful. ↑	Robustness ↑		
			Agg. ↑	Quant. ↑	Relearn ↑
Extraction Strength	**0.85**	0.92	**0.79**	**0.95**	0.68
Exact Mem.	<u>0.80</u>	0.90	0.72	0.92	0.59
Truth Ratio	0.73	**0.95**	0.59	0.92	0.43
Para. Prob.	0.73	0.71	<u>0.75</u>	0.60	**0.98**
Para. ROUGE	0.72	0.89	0.61	<u>0.93</u>	0.45
Probability	0.72	0.82	0.65	0.60	<u>0.70</u>
ROUGE	0.70	0.79	0.64	<u>0.93</u>	0.48
Jailbreak ROUGE	0.69	0.83	0.59	0.85	0.45
MIA - ZLib	0.71	0.92	0.57	0.56	0.59
MIA - MinK	0.67	<u>0.93</u>	0.52	0.48	0.57
MIA - LOSS	0.66	<u>0.93</u>	0.52	0.48	0.57
MIA - MinK++	0.61	0.81	0.48	0.61	0.40

Our extensive testbed of 450+ models supports the development of unlearning metrics that generalize to real-world scenarios. Our goal is to foster more faithful and trustworthy metrics through insights from our meta-evaluation framework. As metrics improve, the testbed should evolve with new unlearning setups, model architectures. Newer adversarial model setups will be needed to challenge metrics as they improve on existing testbeds. Such a dynamic approach ensures that unlearning methods and their meta-evaluations can mutually inform each other, driving progress as unlearning research advances.

4. Conclusion

The field of LLM unlearning has faced challenges due to fragmented methodologies and inconsistent evaluations. To address this, we introduced `OpenUnlearning`, a unified, extensible framework that integrates unlearning algorithms, metrics, and benchmarks. This comprehensive platform enabled us to conduct a novel meta-evaluation of unlearning metrics, assessing their faithfulness and robustness. Our meta-evaluation identified Extraction Strength (ES) and Exact Memorization (EM) as particularly reliable metrics, with Truth Ratio also showing high faithfulness. By standardizing the pipeline and releasing model checkpoints, `OpenUnlearning` paves the way for more rigorous, reproducible, and accelerated progress toward safer unlearning techniques.

References

Abdin, M., Aneja, J., Awadalla, H., Awadallah, A., Awan, A. A., Bach, N., Bahree, A., Bakhtiari, A., Bao, J., Behl, H., et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Bhaila, K., Van, M.-H., and Wu, X. Soft prompting for unlearning in large language models. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4046–4056, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL https://aclanthology.org/2025.naacl-long.204/.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., and Zhang, C. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TatRHT_1cK.

Choi, M., Rim, D., Lee, D., and Choo, J. Opt-out: Investigating entity-level unlearning for large language models via optimal transport, 2024. URL https://arxiv.org/abs/2406.12329.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. QLoRA: Efficient finetuning of quantized llms. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1feb87871436031bdc0f2beaa62a049b-Paper-Conference.pdf.

Ding, C., Wu, J., Yuan, Y., Lu, J., Zhang, K., Su, A., Wang, X., and He, X. Unified parameter-efficient unlearning for LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zONMuIVCAT.

Dong, Y. R., Lin, H., Belkin, M., Huerta, R., and Vulić, I. UNDIAL: Self-distillation with adjusted logits for robust unlearning in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8827–8840, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL https://aclanthology.org/2025.naacl-long.444/.

Doshi, J. and Stickland, A. C. Does unlearning truly unlearn? A black box evaluation of LLM unlearning methods. *arXiv preprint arXiv:2411.12103*, 2024.

Duan, M., Suri, A., Mireshghallah, N., Min, S., Shi, W., Zettlemoyer, L., Tsvetkov, Y., Choi, Y., Evans, D., and Hajishirzi, H. Do membership inference attacks work on large language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=av0D19pSkU.

Eldan, R. and Russinovich, M. Who's harry potter? approximate unlearning in LLMs. *arXiv preprint arXiv:2310.02238*, 2023.

Fan, C., Liu, J., Lin, L., Jia, J., Zhang, R., Mei, S., and Liu, S. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Neurips Safe Generative AI Workshop 2024*, 2024. URL https://openreview.net/forum?id=pVACX02m0p.

Fan, C., Jia, J., Zhang, Y., Ramakrishna, A., Hong, M., and Liu, S. Towards llm unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. *arXiv preprint arXiv:2502.05374*, 2025.

Gandikota, R., Feucht, S., Marks, S., and Bau, D. Erasing conceptual knowledge from language models. *arXiv preprint arXiv:2410.02760*, 2024.

Gao, C., Wang, L., Weng, C., Wang, X., and Zhu, Q. Practical unlearning for large language models. *arXiv preprint arXiv:2407.10223*, 2024a.

Gao, L., Niu, Y., Tang, T., Avestimehr, S., and Annavaram, M. Ethos: Rectifying language models in orthogonal parameter space. *arXiv preprint arXiv:2403.08994*, 2024b.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. The Language Model Evaluation Harness, 07 2024c. URL https://zenodo.org/records/12608602.

Gemma Team, Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Gu, T., Huang, K., Luo, R., Yao, Y., Yang, Y., Teng, Y., and Wang, Y. Meow: Memory supervised llm unlearning via inverted facts, 2024. URL https://arxiv.org/abs/2409.11844.

Gugger, S., Debut, L., Wolf, T., Schmid, P., Mueller, Z., Mangrulkar, S., Sun, M., and Bossan, B. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Hu, S., Fu, Y., Wu, S., and Smith, V. Unlearning or obfuscating? Jogging the memory of unlearned LLMs via benign relearning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fMNRYBvcQN.

Jacobs, S. A., Tanaka, M., Zhang, C., Zhang, M., Song, L., Rajbhandari, S., and He, Y. DeepSpeed Ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023. URL http://arxiv.org/abs/2309.14509.

Jia, J., Zhang, Y., Zhang, Y., Liu, J., Runwal, B., Diffenderfer, J., Kailkhura, B., and Liu, S. Soul: Unlocking the power of second-order optimization for LLM unlearning. *arXiv preprint arXiv:2404.18239*, 2024.

Jin, Z., Cao, P., Wang, C., He, Z., Yuan, H., Li, J., Chen, Y., Liu, K., and Zhao, J. RWKU: Benchmarking real-world knowledge unlearning for large language models. *arXiv preprint arXiv:2406.10890*, 2024.

Karamolegkou, A., Li, J., Zhou, L., and Søgaard, A. Copyright violations and large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=YokfK5VOoz.

Kim, Y., Cha, S., and Kim, D. Are we truly forgetting? A critical re-examination of machine unlearning evaluation protocols. *arXiv preprint arXiv:2503.06991*, 2025.

Li, N., Pan, A., Gopal, A., Yue, S., Berrios, D., Gatti, A., Li, J. D., Dombrowski, A.-K., Goel, S., Phan, L., et al. The WMDP benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

Li, Y., Bubeck, S., Eldan, R., Giorno, A. D., Gunasekar, S., and Lee, Y. T. Textbooks Are All You Need II: Phi-1.5 technical report, 2023. URL https://arxiv.org/abs/2309.05463.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Liu, S., Yao, Y., Jia, J., Casper, S., Baracaldo, N., Hase, P., Xu, X., Yao, Y., Li, H., Varshney, K. R., et al. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787*, 2024.

Łucki, J., Wei, B., Huang, Y., Henderson, P., Tramèr, F., and Rando, J. An adversarial perspective on machine unlearning for AI safety. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=J5IRyTKZ9s.

Lynch, A., Guo, P., Ewart, A., Casper, S., and Hadfield-Menell, D. Eight methods to evaluate robust unlearning in LLMs. *arXiv preprint arXiv:2402.16835*, 2024.

Maini, P., Feng, Z., Schwarzschild, A., Lipton, Z. C., and Kolter, J. Z. TOFU: A task of fictitious unlearning for LLMs. *First Conference On Language Modeling*, 2024. URL https://openreview.net/pdf?id=B41hNBoWLo.

Mekala, A., Dorna, V., Dubey, S., Lalwani, A., Koleczek, D., Rungta, M., Hasan, S., and Lobo, E. Alternate preference optimization for unlearning factual knowledge in large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 3732–3752, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.252/.

Muresanu, A., Thudi, A., Zhang, M. R., and Papernot, N. Unlearnable algorithms for in-context learning, 2024. URL https://arxiv.org/abs/2402.00751.

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

OAG, C. CCPA regulations: Final regulation text. *Office of the Attorney General, California Department of Justice*, 2021.

Qiu, X., Shen, W. F., Chen, Y., Cancedda, N., Stenetorp, P., and Lane, N. D. PISTOL: Dataset compilation pipeline for structural unlearning of LLMs. *arXiv preprint arXiv:2406.16810*, 2024.

Ramakrishna, A., Wan, Y., Jin, X., Chang, K.-W., Bu, Z., Vinzamuri, B., Cevher, V., Hong, M., and Gupta, R. LUME: LLM unlearning with multitask evaluations. *arXiv preprint arXiv:2502.15097*, 2025a.

Ramakrishna, A., Wan, Y., Jin, X., Chang, K.-W., Bu, Z., Vinzamuri, B., Cevher, V., Hong, M., and Gupta, R. SemEval-2025 Task 4: Unlearning sensitive content from large language models. *arXiv preprint arXiv:2504.02883*, 2025b.

Scholten, Y., Günnemann, S., and Schwinn, L. A probabilistic perspective on unlearning and alignment for large language models. *arXiv preprint arXiv:2410.03523*, 2024.

Schwarzschild, A., Feng, Z., Maini, P., Lipton, Z. C., and Kolter, J. Z. Rethinking llm memorization through the lens of adversarial compression. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 56244–56267. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/66453d578afae006252d2ea090e151c9-Paper-Conference.pdf.

Seyitoğlu, A., Kuvshinov, A., Schwinn, L., and Günnemann, S. Extracting unlearned information from LLMs with activation steering. *arXiv preprint arXiv:2411.02631*, 2024.

Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., and Zettlemoyer, L. Detecting pretraining data from large language models, 2023.

Shi, W., Lee, J., Huang, Y., Malladi, S., Zhao, J., Holtzman, A., Liu, D., Zettlemoyer, L., Smith, N. A., and Zhang, C. MUSE: Machine unlearning six-way evaluation for language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TArmA033BU.

Shostack, A. The boy who survived: Removing Harry Potter from an LLM is harder than reported. *arXiv preprint arXiv:2403.12082*, 2024.

Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Team, N., Adler, B., Agarwal, N., Aithal, A., Anh, D. H., Bhattacharya, P., Brundyn, A., Casper, J., Catanzaro, B., Clay, S., Cohen, J., et al. Nemotron-4 340B technical report. *arXiv preprint arXiv:2406.11704*, 2024.

Thaker, P., Hu, S., Kale, N., Maurya, Y., Wu, Z. S., and Smith, V. Position: LLM unlearning benchmarks are weak measures of progress. In *Proceedings of the 3rd IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2025.

Tian, B., Liang, X., Cheng, S., Liu, Q., Wang, M., Sui, D., Chen, X., Chen, H., and Zhang, N. To forget or not? towards practical knowledge unlearning for large language models. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1524–1537, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.82. URL https://aclanthology.org/2024.findings-emnlp.82/.

Tirumala, K., Markosyan, A., Zettlemoyer, L., and Aghajanyan, A. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35: 38274–38290, 2022.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tunstall, L., Beeching, E. E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., Werra, L. V., Fourrier, C., Habib, N., Sarrazin, N., Sanseviero, O., Rush, A. M., and Wolf, T. Zephyr: Direct distillation of LM alignment. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aKkAwZB6JV.

Union, E. Regulation (EU) 2016/679 of the European Parliament and of the Council. *Official Journal of the European Union*, 2016.

Wang, J. G., Wang, J., Li, M., and Neel, S. Pandora's white-box: Precise training data detection and extraction in large language models, 2024. URL https://arxiv.org/abs/2402.17012.

Wang, Q., Han, B., Yang, P., Zhu, J., Liu, T., and Sugiyama, M. Towards effective evaluations and comparisons for LLM unlearning methods. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=wUtCieKuQU.

Wang, Y., Wang, Q., Liu, F., Huang, W., Du, Y., Du, X., and Han, B. GRU: Mitigating the trade-off between unlearning and retention for large language models. *arXiv preprint arXiv:2503.09117*, 2025b.

Wang, Y., Wei, J., Liu, C. Y., Pang, J., Liu, Q., Shah, A., Bao, Y., Liu, Y., and Wei, W. LLM unlearning via loss adjustment with only forget data. In *The Thirteenth International Conference on Learning Representations*, 2025c. URL https://openreview.net/forum?id=6ESRicalFE.

Wei, A., Haghtalab, N., and Steinhardt, J. Jailbroken: How does LLM safety training fail? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=jA235JGM09.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Xu, H., Zhao, N., Yang, L., Zhao, S., Deng, S., Wang, M., Hooi, B., Oo, N., Chen, H., and Zhang, N. Relearn: Unlearning via learning for large language models. *arXiv preprint arXiv:2502.11190*, 2025.

Yadan, O. Hydra - A framework for elegantly configuring complex applications. Github, 2019. URL https://github.com/facebookresearch/hydra.

Yang, B. CE-U: Cross Entropy unlearning. *arXiv preprint arXiv:2503.01224*, 2025.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268–282. IEEE, 2018.

Zhang, J., Sun, J., Yeats, E., Ouyang, Y., Kuo, M., Zhang, J., Yang, H. F., and Li, H. Min-K%++: Improved baseline for pre-training data detection from large language models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=ZGkfoufDaU.

Zhang, R., Lin, L., Bai, Y., and Mei, S. Negative preference optimization: From catastrophic collapse to effective unlearning. *First Conference on Language Modelling*, 2024. URL https://openreview.net/pdf?id=MXLBXjQkmb.

Zhang, Z., Wang, F., Li, X., Wu, Z., Tang, X., Liu, H., He, Q., Yin, W., and Wang, S. Catastrophic failure of LLM unlearning via quantization. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=lHSeDYamnz.

Appendix

A. Limitations

We also note some limitations of our framework and analysis. Firstly, it is limited by the existing popular benchmarks its supports, which have been regarded as "weak measures of unlearning progress" (Thaker et al., 2025). The setups may not accurately reflect realistic model learning or unlearning dynamics, with the underlying forget-retain paradigm itself warranting further scrutiny (Thaker et al., 2025). There's a clear need for more realistic, yet controlled, fine-grained unlearning benchmark setups beyond the currently popular benchmarks. Secondly, while our meta-evaluation of metrics and comparison of methods is a valuable step, its findings need to be extended to more unlearning setups and unlearning algorithms, to gain a greater understanding of the best and comprehensive ways to quantify unlearning. Finally, while our meta-evaluation focuses on knowledge faithfulness and metric robustness as minimal desiderata, these might not be a comprehensive set of desiderata for good unlearning metrics.

B. Broader Impact

The widespread deployment of AI systems in domains ranging from conversational assistants and recommendation systems to self-driving vehicles and medical diagnostics raises important concerns about privacy, safety, and regulatory compliance. As these systems are deeply integrated within society, the ability to remove unwanted or sensitive information from deployed models ("unlearning") is essential to maintain safety, reliability and uphold legal requirements.

Our work on a unified, extensible LLM unlearning benchmark accelerates progress toward reliable, scalable unlearning solutions. By standardizing implementations of unlearning methods, evaluation metrics, and stress tests across diverse tasks and datasets, we lower the barrier for both academic and industrial adoption. This facilitates rapid iteration on novel techniques, ensures consistent measurement of privacy and utility trade-offs, and enables model governance workflows that can respond promptly to deletion or correction requests.

In the long run, advances enabled by this framework will support trustworthy AI deployment in safety-critical and highly regulated settings. From ensuring that autonomous vehicles do not retain outdated or hazardous driving data, to empowering personalized assistants with user-controlled memory, robust unlearning mechanisms will be a cornerstone of ethical, privacy-preserving machine learning. By fostering community collaboration and transparent evaluation, our research paves the way for AI systems that adapt responsibly to evolving societal norms and regulatory landscapes.

C. Overview of LLM unlearning

`OpenUnlearning` uses a common definition of LLM unlearning, where the goal is to eliminate the influence of specific target data or associated model capabilities (Liu et al., 2024), denoted as the "forget set" ($\mathcal{D}_{\text{forget}}$), from an LLM f_{target}. The process pursues two primary goals: (i) *Removal*, ensuring influence caused only by $\mathcal{D}_{\text{forget}}$ is substantially erased, and (ii) *Retention*, maintaining the LLM's utility on downstream tasks not directly linked to $\mathcal{D}_{\text{forget}}$. The setup usually also involves a retain set disjoint from the forget set, used to aid and assess performance preservation.

Formally, given an original model f_{target} trained on a dataset containing $\mathcal{D}_{\text{forget}}$, the unlearning process yields an unlearned model f_{unlearn}. The efficacy of unlearning is typically assessed using evaluation metrics, M, which quantify the remaining influence of $\mathcal{D}_{\text{forget}}$ on f_{unlearn} (e.g., by computing $M(f_{\text{unlearn}}, \mathcal{D}_{\text{forget}})$). Concurrently, utility metrics are used to measure the model's performance on general tasks and data outside of $\mathcal{D}_{\text{forget}}$, ensuring its overall capabilities are preserved.

Unlearning methods: Some LLM unlearning approaches are prompting-based, detecting sensitive queries at inference time and deploying obfuscation mechanisms (Bhaila et al., 2025; Muresanu et al., 2024; Gao et al., 2024a). But these are not practically scalable. Of greater interest is the removal of the forget set's influence directly from the weights. The techniques involved include finetuning with one or more of: (1) tailored loss functions (Maini et al., 2024; Fan et al., 2024; Zhang et al., 2024; Dong et al., 2025; Mekala et al., 2025), (2) optimization modifications (Jia et al., 2024; Wang et al., 2025b; Fan et al., 2025), (3) localized parameter updates (Li et al., 2024; Ding et al., 2025; Gao et al., 2024b), and (4) data based approaches (Maini et al., 2024; Mekala et al., 2025; Xu et al., 2025; Jin et al., 2024; Choi et al., 2024; Gu et al., 2024).

Benchmarks: *Fine-grained unlearning* typically focuses on erasing influence of specific training instances from a forget set while preserving performance on related instances not present in the forget set. TOFU (Maini et al., 2024) introduces fine-grained *knowledge* unlearning using QA-style data from 200 fictitious authors. KnowUndo (Tian et al., 2024) incorporates copyright and privacy aspects through datasets of books and synthetic author profiles. LUME (Ramakrishna et al., 2025a) focuses on unlearning sensitive data from novels, biographies, and real-world figures, emphasizing forgetting of PII. PISTOL (Qiu et al., 2024) builds on TOFU with added structural relationships to study the effect of entity connectivity on knowledge unlearning. MUSE (Shi et al., 2025) also requires fine-grained unlearning, aiming to remove both knowledge, memorization and privacy influence of news articles and copyrighted books. *Open-ended unlearning* can involve a safety-alignment focus as in WMDP (Li et al., 2024) which targets undesired behaviors from hazardous knowledge related to curated datasets. On the other hand, RWKU (Jin et al., 2024) and *Who's Harry Potter* (WHP) task (Eldan & Russinovich, 2023) require forgetting all knowledge related to an entity, without any access to specific retain/forget target corpora. Broadly, benchmarks like TOFU, MUSE, PISTOL, LUME, and KnowUndo involve creating task models by injecting new knowledge via finetuning, while WMDP, RWKU and WHP (Eldan & Russinovich, 2023) operate directly on off-the-shelf LLMs to unlearn existing knowledge.

Unlearning evaluations: Each benchmark task involves multiple evaluations metrics that judge for unlearning success and for general utility preservation. These range from simple probability judgements in TOFU, to MIA-attack based metrics in MUSE, with dozens of metrics across benchmarks in the literature. Evaluating unlearning success is difficult, with several subsequent works questioning the reliability of benchmark metrics in various aspects (Kim et al., 2025; Lynch et al., 2024; Wang et al., 2024; Doshi & Stickland, 2024; Zhang et al., 2025b).

D. Open Unlearning

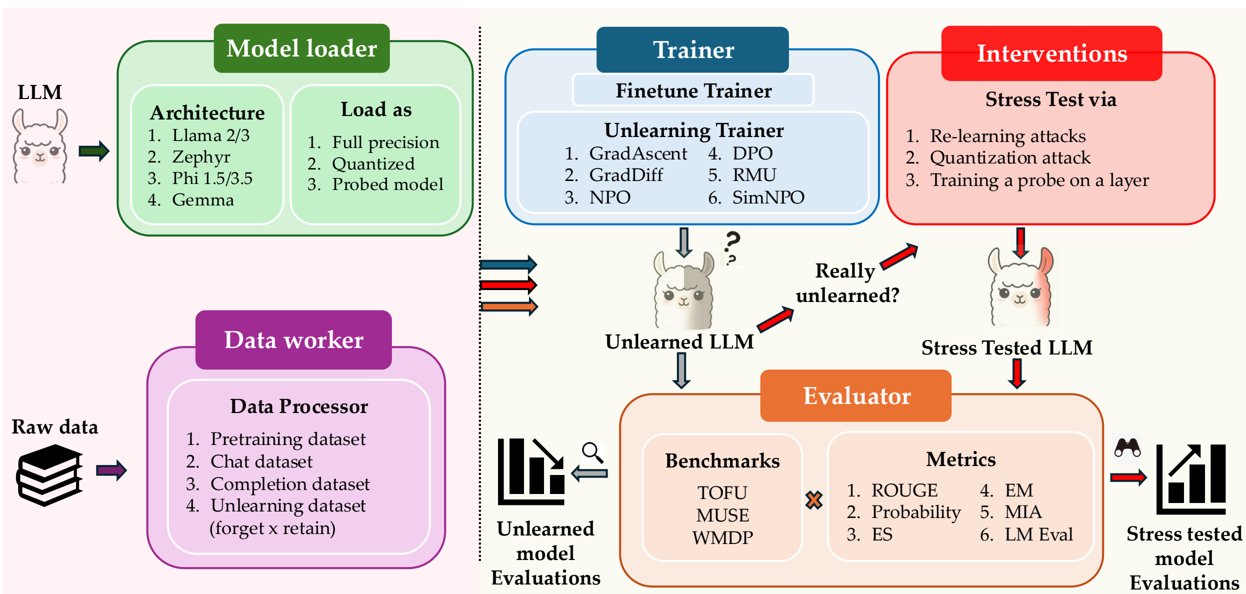

Figure 3. This figure illustrates the unlearning pipeline in terms of implementation-level components. The `OpenUnlearning` framework supports: integration of diverse benchmarks and evaluation suites, implementation of diverse unlearning methods via a unified trainer, evaluation suites, and stress-testing tools to verify unlearning robustness.

D.1. Design of OpenUnlearning

Figure 3 gives an overview of `OpenUnlearning`'s components. Our framework is designed with ease-of-use and easy extensibility in mind. All features are implemented in a structured, modular fashion, simplifying the process for researchers to integrate new datasets, evaluation metrics, unlearning methods, and entire benchmarks. Hydra (Yadan, 2019) is used for configuration management, with `YAML` files specifying each pipeline component and experiment parameters. This helps users effortlessly swap in modules and easily launch an experiment with a single command. A variety of modules, including model-loaders, trainers, dataset preprocessors, evaluation suites, evaluation metrics, experiment types and stress-test interventions are joined together to create the `OpenUnlearning` pipeline (Table 2).

Table 2. OpenUnlearning components and available feature variants in each.

Component		Variants
Models		LLAMA-2, 3.1, 3.2 (Touvron et al., 2023; Grattafiori et al., 2024); ZEPHYR-7B (Tunstall et al., 2024); PHI-1.5, 3.5 (Li et al., 2023; Abdin et al., 2024)
Unlearning algorithms		GradAscent, GradDiff, IdkDPO, IdkNLL (Maini et al., 2024); NPO (Zhang et al., 2024); SimNPO (Fan et al., 2024); RMU (Li et al., 2024); UNDIAL (Dong et al., 2025); AltPO (Mekala et al., 2025)
Datasets		TOFU: bios (Maini et al., 2024); WMDP: cyber, bio (Li et al., 2024); MUSE: news, books (Shi et al., 2025)
Evaluation suites		TOFU (Maini et al., 2024); MUSE (Shi et al., 2025); WMDP (Li et al., 2024); LM Eval (Gao et al., 2024c)
Metrics	**Mem.**	Verbatim Prob. / ROUGE (Maini et al., 2024; Shi et al., 2025); Knowledge QA- ROUGE (Maini et al., 2024; Shi et al., 2025); Extraction Strength (Carlini et al., 2021); Exact Memorization (Tirumala et al., 2022)
	Privacy	Forget Quality (Maini et al., 2024); LOSS (Yeom et al., 2018); ZLib (Carlini et al., 2021); GradNorm (Wang et al., 2024); MinK (Shi et al., 2023); MinK++ (Zhang et al., 2025a); Privacy Leakage (Shi et al., 2025)
	Utility	Truth Ratio, Model Utility (Maini et al., 2024); LM-Eval (Gao et al., 2024c) (WMDP, MMLU, etc.); Fluency (Mekala et al., 2025)
Stress tests		Relearning (Hu et al., 2025; Lynch et al., 2024; Łucki et al., 2025; Wang et al., 2025a); Quantization (Zhang et al., 2025b); Probing (Lynch et al., 2024; Seyitoğlu et al., 2024; Wang et al., 2025a)

D.2. Design of modules

(a) Method implementation leveraging HuggingFace Trainer, followed by registration.

```python
from transformers import Trainer
class Unlearner(Trainer):
  def compute_loss(self, ...):
    ...
  def get_optimizer_cls_and_kwargs(...):
    # custom optimizer
    ...
  def _inner_training_loop(self, ...):
    # modify training logic
    ...
_register_trainer(Unlearner)
```

(b) Configuration: create a YAML config specifying Training args and method parameters.

```yaml
handler: Unlearner # map registered name

args: # HuggingFace Trainer args
  num_epochs: 10
  learning_rate: 1e-5
  optim: shampoo

method_args:
  alpha: 1.0
  switch_every_n: 10
  retain_loss_type: NLL
```

Figure 4. Illustration of implementing a hypothetical unlearning method in `OpenUnlearning`

The procedure of extending `OpenUnlearning` with a new module variant generally involves two simple steps. (1) **Create and register a handler.** The `Python` class or function encapsulating the component's logic is implemented then registered to be accessed via a string key. (2) **Create the config.** The configuration `YAML` file names the handler key and specifies its parameters. Figure 4 provides an example illustrating this procedure for a new unlearning method.

Features: We currently support 9 unlearning algorithms, 6 model architectures, and 5 datasets ranging from chat to pretraining. Among existing benchmarks, we focus on the three most cited and used TOFU (Maini et al., 2024), MUSE (Shi et al., 2025), WMDP (Li et al., 2024) benchmarks. The framework includes a diverse set of metrics to assess model performance, including 16 unlearning metrics from existing benchmarks, as well as additional evaluations by integrating

LM Eval Harness (Gao et al., 2024c). We also support three stress-testing approaches, which are essential for testing the robustness of unlearning, usually critical for model-owners in verifying compliance. All these features are summarized in Table 2 by component and variant. Our integration enriches each benchmark by enabling the use of metrics originally developed for others. For example, PrivLeak, initially introduced in MUSE, is now available in TOFU. More details on these technical benchmark improvements can be found in Appendix D.3. We also encourage community contributions by providing detailed guidelines for adding new benchmarks, unlearning methods, and evaluation metrics. This has already resulted in contributions from the community, with implementations for works like (Dong et al., 2025; Wang et al., 2025b; Yang, 2025).

`OpenUnlearning` is a living framework, and our design choices are built keeping easy integration of new components in mind. For instance, since the public release of our repository (with just TOFU and MUSE benchmarks) we introduced the WMDP benchmark, unlearning methods like RMU (Li et al., 2024), UNDIAL (Dong et al., 2025), AltPO (Mekala et al., 2025); evaluations like ES (Carlini et al., 2021), EM (Tirumala et al., 2022), MIA (Duan et al., 2024) and integrated evaluations like MUSE's PrivLeak (into TOFU) and LM Eval Harness (Gao et al., 2024c) (to enable WMDP evaluation) among many others. Additionally, we encourage community contributions by providing detailed guidelines for adding new benchmarks, unlearning methods, and evaluation metrics. This has already resulted in contributions from the community, with implementations for works like (Dong et al., 2025; Wang et al., 2025b; Yang, 2025). Currently, each module supports several variants, with 3 popular LLM unlearning benchmarks, 5 task datasets, 9 unlearning methods, 16 evaluation metrics, 6 LLM architectures and 3 stress-tests.

D.3. Unlearning benchmarks

TOFU: A synthetic fine-grained knowledge-unlearning benchmark with 200 fictitious author profiles, each offering 20 QA pairs and a defined "forget set", and a finetuned chat LLM. TOFU's primary metric is Truth Ratio, which measures the *relative* likelihood of the true answer after unlearning.

MUSE: A memorization and knowledge unlearning benchmark targeting the removal of books and news articles from a finetuned LLM. MUSE evaluates for memorization (via verbatim reproduction rates), knowledge (via question-answers) and privacy protection (using membership-inference attacks).

WMDP: An alignment-focused benchmark of 3,668 multiple-choice questions probing hazardous knowledge in biosecurity, cybersecurity, and chemical security, paired with corresponding unlearning corpora and off-the-shelf chat LLMs. WMDP assesses a model's ability to forget dangerous capabilities while preserving general performance.

Improvements: In evaluations, TOFU reuses training questions, raising concerns about overfitting and inflated metrics. To mitigate this, we evaluate on paraphrased questions in our meta-evaluation and benchmarking. We also extend TOFU with privacy-based metrics from MUSE via PrivLeak (Shi et al., 2025) and introduce additional MIA attacks. For this we create new holdout datasets by replicating the original TOFU data generation setup.[1] We add MIA beyond Min-K (Shi et al., 2023) to MUSE. Given the poor quality and tokenization issues users faced with the PHI-1.5 and LLAMA-2 models from TOFU, we introduce new starter target models. `OpenUnlearning` provides three sizes of the recent LLAMA-3 models: 1B, 3B, and 8B, giving users greater flexibility to experiment. Additionally, we augment both TOFU and MUSE with metrics such as Extraction Strength (Carlini et al., 2021), Exact Memorization (Tirumala et al., 2022), and Forget Fluency (Mekala et al., 2025). We integrate `OpenUnlearning` with LM Eval Harness (Gao et al., 2024c) to assess general LLM capabilities that identify post-unlearning degradations, in addition to enabling WMDP evaluations. Several contemporary works can further enhance these benchmarks. We plan to continuously improve the framework by adding-and encouraging contributions of-new features and metrics to both existing and future benchmarks, such as the recent work by Thaker et al. (2025).

D.4. Datasets

In machine unlearning, benchmarks typically structure data into two primary components: (1) forget sets, which contain text corpora and queries designed to test whether the model has successfully erased targeted information, and (2) retain sets, which verify that the model preserves unrelated, desirable knowledge. Beyond this fundamental split, unlearning benchmarks often include additional variations to test algorithmic robustness. For example, scaling splits vary the size of the forget set to assess how well algorithms handle larger deletion requests, while topic-based splits examine whether forgetting specific content impacts retention across semantically related or unrelated domains (Maini et al., 2024; Shi et al., 2025).

[1] We use the same `gpt-4-1106-preview` endpoint and prompts for data generation.

These nuanced splits are essential for assessing scalability, generalization, and sustainability of unlearning methods under realistic conditions.

(a) Dataset Handler

```python
class PretrainingDataset(Dataset):
    def __init__(self, hf_args, ...):
        ...

    def __getitem__(self, idx):
        ...
        return item

_register_data(PretrainingDataset)
```

(b) Dataset Configuration

```yaml
MUSE_forget:
  handler: PretrainingDataset
  args:
    hf_args:
      path: "muse-bench/MUSE-News"
      name: "raw"
      split: "forget"
    text_key: "text"
    max_length: 2048
```

Figure 5. Adding a dataset in `OpenUnlearning`: (a) the Python handler class implementing data preprocessing and reusable to load several datasets, and (b) the configuration file specifying arguments for instantiating a particular dataset variant. Adding variants of other modules (e.g. unlearning method trainers, benchmarks, evaluation metrics etc.) involves a similar procedure.

`OpenUnlearning` provides a modular framework where most of the Python implementation for dataset classes is shared across various dataset configurations and benchmarks. It also allows users to define custom dataset classes following the steps presented in Figure 5. We already support three commonly used dataset handlers, each serving a distinct purpose in the unlearning pipeline:

- `PretrainingDataset`: used for training models on large-scale web corpora. This handler is essential for simulating pretraining kind of settings.
- `CompletionDataset`: used for evaluating model outputs in a zero-shot or few-shot setting. This format is particularly useful for measuring memorization and information leakage, such as verbatim reproduction of forgotten content.
- `QADataset`: designed for probing models using natural language question-answer interactions, optionally with few-shot examples. This format is critical for assessing whether the model retains or forgets factual knowledge in interactive settings. Moreover, the framework automatically pipelines model-specific input formatting such as including system prompts or special tokens for chat-based models ensuring that queries are executed in a manner consistent with the model's native interface.
- `ForgetRetainDataset`: The unlearning process involves simultaneous optimization on both the forget and retain datasets, requiring concurrent batch loading. This dataset class abstracts this by loading the retain dataset in the same order as the forget dataset for unlearning.

D.5. Metrics

`OpenUnlearning` supports multiple evaluation metrics and shares common functionalities across metric implementations. Metrics are broadly classified into three categories and summarized below:

Memorization Metrics: These metrics quantify how much the model has memorized information from its training data.

1. **Exact Memorization (EM):** Quantifies memorization by calculating proportion of tokens in the model's response that exactly match those in the ground truth y (Tirumala et al., 2022). Formally, it is defined as

$$\text{EM} = \frac{1}{|y|} \sum_{k} \mathbf{1} \left\{ \arg\max_{y} f(y \mid [x, y^{<k}]; \boldsymbol{\theta}) = y^{k} \right\}, \tag{5}$$

2. **Extraction Strength (ES):** Quantifies the intensity of memorization by determining the minimal prefix length required to reconstruct the remaining suffix (Carlini et al., 2021).

$$\text{ES} = 1 - \frac{1}{|y|} \min_{k} \left\{ k \mid f([x, y^{<k}]; \boldsymbol{\theta}) = y^{>k} \right\}. \tag{6}$$

3. **Probability (Prob.):** Directly quantifies the model's confidence in its output.

$$\text{Probability} = p(f(y|x) \tag{7}$$

4. **Paraphrased Probability (Prob.):** Probability is computed on a paraphrased y^{para} instead of y to remove original answer template bias. The goal is to capture the knowledge rather than surface-level lexical overlap.

$$\text{Para. Prob.} = p(f(y^{\text{para}}|x) \quad (8)$$

5. **ROUGE/Paraphrased ROUGE:** Assesses the degree of overlap between the model's output $f(x)$ and the ground truth y (Lin, 2004). This can be computed against many variants of datasets, including paraphrases and jailbreak prompts (next).
6. **Jailbreak ROUGE:** To probe for forgotten information, we employ a prefix-based jailbreaking attack by prompting the model with *"Sure, here is the answer:"* (as in (Wang et al., 2025a)) and then computing the ROUGE score between the model's response and the ground truth. This metric captures the extent to which suppressed content can still be recovered through prompt manipulation.
7. **Truth Ratio:** Measures the model's preference for the correct answer over a perturbed (incorrect) alternative by comparing their predicted probabilities. A higher value indicates stronger confidence in the correct response. It is defined as:

$$\text{Truth Ratio} = \frac{p(y^{\text{para}} \mid x)}{p(y^{\text{para}} \mid x) + p(y^{\text{pert}} \mid x)} \quad (9)$$

where y_{para} denotes the paraphrased correct answer and y_{pert} represents a incorrect alternative keeping the sentence structure similar. Note that Maini et al. (2024) use a privacy-oriented variant of Truth Ratio computed as Truth Ratio $=$ $min(\frac{p(y^{\text{para}}|x)}{p(y^{\text{pert}}|x)}, \frac{p(y^{\text{pert}}|x)}{p(y^{\text{para}}|x)})$. We modify it so that it quantifies extent of knowledge for our work's purposes.

Privacy Metrics: These metrics ascertain whether sensitive information from the forget set can still be inferred or extracted from the model. Techniques such as Membership Inference Attacks (MIA) are utilized to evaluate the model's susceptibility to revealing whether specific data points were part of its training set, thereby assessing the privacy guarantees post-unlearning. However, these metrics often assumes an access to the perfect i.i.d houldout splits or access to the 'oracle' retain model limiting its practical usage in real-world.

1. **MIA:** Evaluates a model's tendency to memorize training data by testing whether an adversary can distinguish between seen examples from the forget set ($\mathcal{D}_{\text{forget}}$) and unseen examples from a holdout set ($\mathcal{D}_{\text{holdout}}$), based on model confidence. Ideally, a model that has not seen the forget set should yield an AUC of 0.5. However, due to challenges in constructing perfect holdout splits, benchmarks such as MUSE often calibrate this with AUC scores from the retain model (e.g., as done in PrivLeak). We support several MIA methods, including: LOSS (Yeom et al., 2018), ZLib (Carlini et al., 2021), GradNorm (Wang et al., 2024), MinK (Shi et al., 2023), and MinK++ (Zhang et al., 2025a).
2. **Forget Quality:** Performs a statistical test on the truth ratio distributions of the unlearned and retain models, yielding high values when the distributions closely match.

$$\text{KS}(\text{Truth Ratio}(f_{\text{target}}, \mathcal{D}_f), \text{Truth Ratio}(f_{\text{retain}}, \mathcal{D}_f)) \quad (10)$$

Utility Metrics: The goal of unlearning is to effectively forget the targeted data while preserving the model's performance on non-forget data. Utility metrics assess whether the model retains its capabilities on broader tasks beyond the retain data, ensuring that unlearning does not degrade general performance on real-world distributions.

1. **Model Utility (MU):** Captures the retained performance of a model after unlearning, both on the closely tied retain set and on broader general knowledge. TOFU computes MU as the harmonic mean of nine metrics across three data levels: the retain set, real authors, and factual world knowledge. At each level, it evaluates three metrics—probability, ROUGE, and the Truth Ratio.
2. **ROUGE for knowledge:** MUSE, TOFU assess utility by measuring ROUGE on knowledge-based questions.
3. **Forget Fluency:** Prior work (Mekala et al., 2025; Gandikota et al., 2024) has shown that unlearning often degrades model fluency, particularly on the forget set, resulting in random or nonsensical outputs. To capture this effect, we employ a classifier-based score that predicts whether a given text resembles gibberish[2].
4. **LM Eval Harness**: LM Evaluation Harness (Gao et al., 2024c) is an easy to use library enabling evaluations for a wide variety of general LLM benchmarks. It is integrated into `OpenUnlearning`, unlocking a broad suite of metrics such as WMDP MCQ, MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021) etc., for comprehensive post-unlearning evaluation.

[2] https://huggingface.co/madhurjindal/autonlp-Gibberish-Detector-492513457

By integrating the diverse metrics listed in Table 2, `OpenUnlearning` offers a robust framework to holistically evaluate unlearning methods, ensuring that models not only forget specific data but also maintain utility and privacy standards. Figure 6 illustrates the process of adding a new metric to the `OpenUnlearning` framework.

It is important to recognize that the applicability of unlearning metrics often depends on the dataset used during evaluation. As a result, metrics implemented for one benchmark may not directly transfer to another. For example, the Knowledge Memorization metric in MUSE is based on question-answer pairs where answers are typically short, single-word responses. In contrast, TOFU lacks such a data split and instead features more descriptive, verbose answers. In this context, metrics like ROUGE recall may inadvertently capture surface-level template patterns rather than the core semantic content, potentially misleading the evaluation.

(a) Metric Handler

```python
@unlearning_metric(name="rouge")
def rouge(model, **kwargs):
    tokenizer = kwargs["tokenizer"]
    data = kwargs["data"]
    collator = kwargs["collators"]
    batch_size = kwargs["batch_size"]
    generation_args = kwargs["generation_args"]
    ... # calculate ROUGE
    return {
        "agg_value": np.mean(rouges),
        "value_by_index": rouges,
    }
```

(b) Metric Configuration

```yaml
# @package eval.muse.metrics.forget_verbmem_ROUGE
defaults: # fill up forget_verbmem_ROUGE's inputs' configs
  - ../../data/datasets@datasets: MUSE_forget_verbmem
  - ../../collator@collators: DataCollatorForSupervisedDatasetwithIndex
  - ../../generation@generation_args: default
handler: rouge # the handler we defined above in (a)
rouge_type: rougeL_f1
batch_size: 8
datasets:
  MUSE_forget_verbmem:
    args:
      hf_args:
        path: muse-bench/MUSE-Books
      predict_with_generate: True
collators:
  DataCollatorForSupervisedDataset:
    args:
      padding_side: left # for generation
generation_args:
  max_new_tokens: 128
```

Figure 6. Example of a metric definition in `OpenUnlearning`: (a) the Python handler that implements the ROUGE metric, and (b) the corresponding configuration used to run ROUGE-based evaluation for assessing verbatim memorization.

D.6. Models

Different language models encode and store knowledge in fundamentally different ways depending on their architecture and training setup. As a result, evaluating unlearning methods across a diverse range of models is essential for assessing their robustness and generalizability. However, existing benchmark implementations often support only a narrow set of model types and require users to manually rewrite evaluation logic such as input formatting, tokenization, and prompting—when adapting to new architectures. For example, chat-based models rely on specialized prompting structures that differ significantly from standard causal language models, making adaptation tedious and error-prone.

`OpenUnlearning` supports multiple model architectures and sizes out of the box. Built on Hugging Face Transformers (Wolf et al., 2020), it uses `AutoModelForCausalLM` and `AutoTokenizer`, while also supporting custom model loading (e.g., for probe models). A unified abstraction allows seamless switching between chat-style and base models without modifying the unlearning or evaluation pipeline, reducing overhead and enabling consistent cross-model comparisons.

In addition to support loading models in multiple precisions, `OpenUnlearning` also support loading 4-bit and 8-bit quantized models using the `bitsandbytes` library Dettmers et al. (2023). This flexibility for quantization is particularly valuable for stress testing unlearning Zhang et al. (2025b).

Table 3. Supported LLM Architectures in `OpenUnlearning`

Model	Reference
LLAMA-2	Touvron et al. (2023)
LLAMA-3.1 / 3.2	Grattafiori et al. (2024)
PHI-1.5	Li et al. (2023)
PHI-3.5	Abdin et al. (2024)
GEMMA	Gemma Team et al. (2024)
ZEPHYR	Tunstall et al. (2024)

New models for TOFU : `OpenUnlearning` provides trained models for the TOFU benchmark using LLAMA-based architectures finetuned on the TOFU dataset. These models span a range of sizes including 1B, 3B, and 8B parameters, enabling users to explore unlearning behavior across different model capacities. The 1B model, in particular, offers a highly efficient option for rapid experimentation with turnaround time of 15 minutes, requiring only 20 GB of GPU VRAM.

(a) LLAMA 3.2 1B model configuration

```yaml
model_args:
  pretrained_model_name_or_path: "meta-llama/Llama-3.2-1B-Instruct"
  attn_implementation: 'flash_attention_2'
  torch_dtype: bfloat16
tokenizer_args:
  pretrained_model_name_or_path: "meta-llama/Llama-3.2-1B-Instruct"
template_args:
  apply_chat_template: True
  system_prompt: You are a helpful assistant.
  date_string: 10 Apr 2025
```

(b) LLAMA 2-7B model configuration

```yaml
model_args:
  pretrained_model_name_or_path: "meta-llama/Llama-2-7b-hf"
  attn_implementation: 'flash_attention_2'
  torch_dtype: bfloat16
tokenizer_args:
  pretrained_model_name_or_path: "meta-llama/Llama-2-7b-hf"
template_args:
  apply_chat_template: False
  user_start_tag: "Question: "
  user_end_tag: "\n"
  asst_start_tag: "Answer: "
  asst_end_tag: "\n\n"
```

Figure 7. Example model configurations for two different LLAMA variants: (a) LLAMA 3.2-1B with chat template prompting, and (b) LLAMA 2-7B with manual prompt formatting.

D.7. Unlearning Methods

Unlearning methods form the core of the `OpenUnlearning` framework. In practice, researchers proposing new unlearning approaches often evaluate them on a single benchmark due to the high efforts of adapting their code to other frameworks. This fragmentation has led to a lack of comprehensive, cross-benchmark comparisons in the unlearning literature. The overhead of re-implementing methods, adapting to different evaluation pipelines, and aligning metrics discourages reproducibility and slows progress.

`OpenUnlearning` addresses this gap by providing a unified and modular infrastructure that abstracts away benchmark-specific details. Researchers can implement their method once, typically by extending a custom `Trainer`, and instantly evaluate it across multiple benchmarks using a consistent API and evaluation suite. This design dramatically lowers the barrier to rigorous, multi-benchmark evaluation and encourages the community to develop more generalizable and robust methods. We currently support all commonly used baselines as well as several state-of-the-art methods, and we invite the community to build upon this foundation.

Gradient Ascent (Maini et al., 2024): Performs gradient ascent on the forget set to degrade model confidence on targeted data.

$$\mathcal{L} = -\gamma \mathbb{E}_{(x,y_\text{f})\sim\mathcal{D}_\text{forget}} \ell\big(y_\text{f}|x; f_\text{unl}\big) \tag{11}$$

GradDiff (Maini et al., 2024): Performs gradient ascent on forget data and descent on retain data.

$$\mathcal{L} = -\gamma \mathbb{E}_{(x,y_\text{f})\sim\mathcal{D}_\text{forget}} \ell\big(y_\text{f}|x; f_\text{unl}\big) + \alpha \mathbb{E}_{(x,y)\sim\mathcal{D}_\text{retain}} \ell\big(y|x; f_\text{unl}\big)$$

IdkNLL (Maini et al., 2024): Trains to output "I don't know" responses when queried on forgotten content.

$$\mathcal{L} = \gamma \mathbb{E}_{(x,y_\text{f})\sim\mathcal{D}_\text{forget}} \ell\big(y_\text{idk}|x; f_\text{unl}\big) + \alpha \mathbb{E}_{(x,y)\sim\mathcal{D}_\text{retain}} \ell\big(y|x; f_\text{unl}\big)$$

IdkDPO (Maini et al., 2024): Uses a DPO-style objective to align the model to output "I don't know" responses when queried on forgotten content.

$$\begin{aligned}\mathcal{L} = -\frac{2}{\beta} \mathbb{E}_{(x,y_\text{f})\sim\mathcal{D}_\text{forget}} \log\sigma\Big(-\beta\log\Big(\frac{p(y_\text{idk}|x; f_\text{unl})}{p(y_\text{idk}|x; f_\text{target})}\Big) - \beta\log\Big(\frac{p(y_\text{f}|x; f_\text{unl})}{p(y_\text{f}|x; f_\text{target})}\Big)\Big) \\ + \alpha \mathbb{E}_{(x,y)\sim\mathcal{D}_\text{retain}} \ell\big(y|x; f_\text{unl}\big)\end{aligned}$$

NPO (Zhang et al., 2024): Similar to the DPO-style objective, but uses only the negative feedback term in its formulation. It demonstrates better training stability compared to similar methods like GradDiff.

$$\begin{aligned}\mathcal{L} = -\frac{2}{\beta} \mathbb{E}_{(x,y_\text{f})\sim\mathcal{D}_\text{forget}} \log\sigma\Big(-\beta\log\Big(\frac{p(y_\text{f}|x; f_\text{unl})}{p(y_\text{f}|x; f_\text{target})}\Big)\Big) \\ + \alpha \mathbb{E}_{(x,y)\sim\mathcal{D}_\text{retain}} \ell\big(y|x; f_\text{unl}\big)\end{aligned}$$

SimNPO (Fan et al., 2024): A simplified and efficient variant of NPO that retains its core forgetting behavior with by replacing the reference model with δ in loss formulation.

$$\mathcal{L} = -\frac{2}{\beta} \mathbb{E}_{(x,y_\text{f})\sim\mathcal{D}_\text{forget}} \log\sigma\Big(-\frac{\beta}{|y_\text{f}|}\log p(y_\text{f}|x; f_\text{unl}) - \delta\Big)\Big) + \alpha \mathbb{E}_{(x,y)\sim\mathcal{D}_\text{retain}} \ell\big(y|x; f_\text{unl}\big)$$

AltPO (Mekala et al., 2025): Uses a DPO-style objective to align the model toward generating alternate, in-domain plausible facts (produced by the model itself) that introduce ambiguity and suppress the original target knowledge.

$$\begin{aligned}\mathcal{L} = -\frac{2}{\beta} \mathbb{E}_{(x,y_\text{f})\sim\mathcal{D}_\text{forget}} \log\sigma\Big(-\beta\log\Big(\frac{p(y_\text{alt}|x; f_\text{unl})}{p(y_\text{alt}|x; f_\text{target})}\Big) - \beta\log\Big(\frac{p(y_\text{f}|x; f_\text{unl})}{p(y_\text{f}|x; f_\text{target})}\Big)\Big) \\ + \alpha \mathbb{E}_{(x,y)\sim\mathcal{D}_\text{retain}} \ell\big(y|x; f_\text{unl}\big)\end{aligned}$$

RMU (Li et al., 2024): Assumes knowledge is encoded in model parameters and manipulates these representations to suppress memorization signals for the forget set while preserving knowledge in the retain set. Let $\phi(s; f_{\text{unl}})$ denote the embedding features of the model, the loss is given by

$$\begin{aligned}\mathcal{L} =& \mathbb{E}_{(x,y_f)\sim\mathcal{D}_{\text{forget}}} \frac{1}{|y_{\text{f}}|} \sum_{i=1}^{|y_{\text{f}}|} ||\boldsymbol{\phi}([x, y^{<i}]; f_{\text{unl}}) - c \cdot \boldsymbol{u}||_2^2 \\ &+ \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{retain}}} \frac{1}{|y|} \sum_{i=1}^{|y|} ||\boldsymbol{\phi}([x, y^{<i}]; f_{\text{unl}}) - \boldsymbol{\phi}([x, y^{<i}]; f_{\text{target}})||_2^2,\end{aligned}$$

where $\boldsymbol{u}$ has elements randomly sampled from $[0, 1)$ and c is a scaling hyper-parameter.

UNDIAL (Dong et al., 2025): Mitigates the instability found in prior methods by employing self-distillation, where the model learns from its own adjusted outputs. The core idea is to reduce the model's confidence in the target token by adjusting its logits, thereby diminishing its influence without affecting the overall model performance. This is achieved by minimizing the KL divergence between the adjusted logits and the model's current output distribution.

$$\begin{aligned} z_{\text{adj}}(x) &= z_{\text{orig}}(x) - \beta \cdot \mathbf{1}_{y_f} \\ \mathcal{L} &= \gamma \mathbb{E}_{(x,y_f)\sim\mathcal{D}_{\text{forget}}} \left[\text{KL} \left(\text{softmax}(z_{\text{adj}}(x)) \, \| \, \text{softmax}(z_{\text{unl}}(x)) \right) \right] + \alpha \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{retain}}} \ell(y|x; f_{\text{unl}}) \end{aligned}$$

Where $z_{\text{orig}}(x)$ is the original logits produced by the model before unlearning and $z_{\text{adj}}(x)$ is the adjusted logits.

D.8. Technical improvements:

Efficiency: MUSE evaluates models without batching, while our implementation uses batched inference to improve efficiency. TOFU pads all sequences to a fixed `max_length` of 512, resulting in unnecessary GPU memory and compute overhead. In contrast, we apply dynamic padding based on the longest sequence in each batch. WMDP lacks a rigorous training and unlearning framework, limiting its extensibility for developing and evaluating new methods.

Training paradigms supported: Training or unlearning with larger models (e.g., $\geq$ 8B parameters) presents a significant computational challenge, often necessitating multiple high-end GPUs such as NVIDIA A100s. To acclerate this process we support,

1. **DeepSpeed ZeRO Stage-3** (Jacobs et al., 2023): Enabled via the Accelerate library (Gugger et al., 2022), reducing the memory usage through optimizer state partitioning and CPU/NVMe offloading.
2. **Model Parallelism**: Splits the model across GPUs along its layers, allowing large models to be trained even when individual GPUs cannot hold the full model in memory.

E. Experimental setup

All subsequent meta-evaluation and benchmarking experiments use the LLAMA-3.2-1B model. Experiments use BF16 precision, a single NVIDIA A100 GPU, a batch size of 32 and a paged AdamW optimizer (matching the TOFU paper's default settings).

F. Meta-evaluation

F.1. Faithfulness test-bed design

We create two pools of models: the negative N and the positive P pool. N contains models trained with varying training parameters while avoiding the knowledge of the forget set in the training data. P contains models trained similarly to N but with the target knowledge included in training. During the model pool preparation, we modify the training data used in the N and P pools with several training data variants. This introduces model diversity, forcing metrics to detect genuine *knowledge* retention rather than non-knowledge related artifacts, to achieve high scores. The faithfulness evaluation pipeline is illustrated in Figure 1 (a).

1. **Positive pool (P):** Models are trained on all TOFU facts (both `forget10` and `retain90`). We then replace

`forget10` with two transformed variants. First, `forget10_paraphrased` uses paraphrased labels while preserving factual content. Second, `forget10_bio` contains long-form biographies derived from `forget10`.

2. **Negative pool (N):** Models are trained on the `retain90` split of TOFU, along with two perturbed variants of `forget10`. First, `forget10_perturbed` pairs each forget prompt with an incorrect label. Second, `celeb_bio` (biographies of random celebrities) serves as the counterpart to `forget10_bio`.

To further diversify the model pool, we vary training hyperparameters: five learning rates from 1×10^{-5} to 5×10^{-5}, and two checkpoints (after training epochs 5 and 10). Combining 2 pools $\times$ 3 dataset variants $\times$ 5 learning rates $\times$ 2 checkpoints yields 60 models in total.

Data generation process While some of TOFU's evaluation datasets include paraphrased and perturbed examples, our training-set variants for the model pool were generated independently. We used LLAMA 3.1 405B via the SambaNova API[3] to paraphrase and perturb QA pairs, and prompted Gemini[4] to produce Wikipedia-style biographies from each author's 20 QA pairs.

F.2. Robustness setup design

We create a large and diverse pool of unlearned models and a separate set of retain models, which serve as gold-standard references having never been trained on the forget set. The unlearned pool is then subjected to stress-test interventions, to provoke recovery (or inducing) of the forgotten knowledge. These pools serve as our test-bed. For every metric being meta-evaluated, values are recorded on both pools before and after each intervention. The change in a metric's distribution before and after intervention on the unlearned models (along with the change in retain models for normalization) is used to characterize robustness. We use three interventions: *relearning*, *quantization* and *probing*.

1. **Relearning Setup:** We finetune the unlearned model on the full `forget10` dataset for one epoch with a learning rate of 2×10^{-5}.

2. **Quantization Setup:** We apply 4-bit floating-point quantization using BitsAndBytes (Dettmers et al., 2023). Checkpoints unlearned with a learning rate of 1×10^{-5} are chosen, as quantization is most effective at lower learning rates (Zhang et al., 2025b).

3. **Probing:** We evaluate layer 11 of the LLAMA-3.2-1B model (16 layers total) using the language-model head from the corresponding `retain90`-trained model. This head is trained with a learning rate of 1×10^{-4} on `retain90` for ten epochs.

F.3. Additional Results

Figure 8 shows the faithfulness of the metrics, while Figure 9 and Figure 10 show their behavior under relearning and quantization stress tests. We found that removing MU filter of retaining at least 80% utility for unlearned models reduces robustness to quantization further (see Figure 11). Despite this, we apply the MU filter to better align with common unlearning reporting practices.

Probing results: We compute the metric robustness to probing intervention as follows

$$p = \frac{m^a_{\text{ret}}}{m^a_{\text{unl}}} \quad \text{if} \quad \frac{m^b_{\text{ret}}}{m^b_{\text{unl}}} \geq 1, \quad P = \min(p, 1) \tag{12}$$

Table 4 shows the results of our metric meta-evaluation with probing. Probing, while provided for by `OpenUnlearning`, is not used in the meta-evaluation procedure, as P scores on TOFU achieve 1 for all metrics and thus offer little information.

F.4. Further considerations

Why aren't the intervened versions of metrics considered evaluation metrics themselves? The interventions we use require modification to and access of model weights, which an unlearning auditor might not possess. In the case of

[3] https://cloud.sambanova.ai/playground

[4] `gemini-2.0-flash-exp` (accessed 26 April 2025)

Table 4. Robustness meta-evaluation with probing (layer 11)

Metrics	Probe ↑
Exact Mem.	1.0
Extr. Strength	1.0
Truth Ratio	1.0
Prob.	0.99
ROUGE	0.99
Jailbreak ROUGE	0.99
Para. Prob.	1.0
Para. ROUGE	0.99
MIA - LOSS	1.0
MIA - MinK	1.0
MIA - MinK++	0.83
MIA - ZLib	1.0

relearning and quantization, they also involve computational costs associated with training. Stress-testing interventions are best suited for final-stage audits before model deployment, rather than for routine use throughout unlearning workflows, as is expected of standard evaluation metrics. Our analysis can inform the design of robust evaluation metrics that function without requiring stress-testing.

Comparison to Wang et al. (2025a)'s meta-evaluation: Our work is related to the recent effort by Wang et al. (2025a) to compare unlearning evaluation metrics. Their analysis focuses on four metrics: probability, ROUGE, ES, and EM, and evaluates robustness by measuring the linear correlation of metric values before and after applying stress-tests such as jailbreaking, relearning, probing, and token noising. We extend this framework in several key ways.

1. **Broader metric coverage:** We evaluate a broader range of metrics, including six additional ones.

2. **Faithfulness assessment:** We assess faithfulness of metrics in our meta-evaluation as a minimal criterion. This enforces that good metrics must accurately capture the presence or absence of target knowledge, rather than merely resisting change under intervention.

3. **Focused interventions:** We focus specifically on three interventions: relearning, probing, and quantization, excluding jailbreaking and token-noising from the intervention set. We instead treat jailbreaking it as an evaluation metric in its own right. Prompt-based attacks like paraphrasing and jailbreak-style prompts are more naturally seen as inexpensive evaluation metrics rather than stress-testing interventions. Additionally, Wang et al. (2025a) found jailbreaking and token noising (which is also a prompt modification) to be less effective as interventions.

4. **A different calibration criterion:** our procedure also introduces a calibration criterion grounded in ideal behavior. Rather than expecting linear variation from a metric upon intervention, we benchmark metric behavior against a gold-standard retain model, for a more principled signal of robustness.

5. **Practical robustness analysis:** Our robustness analysis filters for models with good utility that are substantially unlearned, selected from a diverse and representative set of unlearning algorithms. This leads to a test distribution for metrics that better reflects realistic unlearning scenarios.

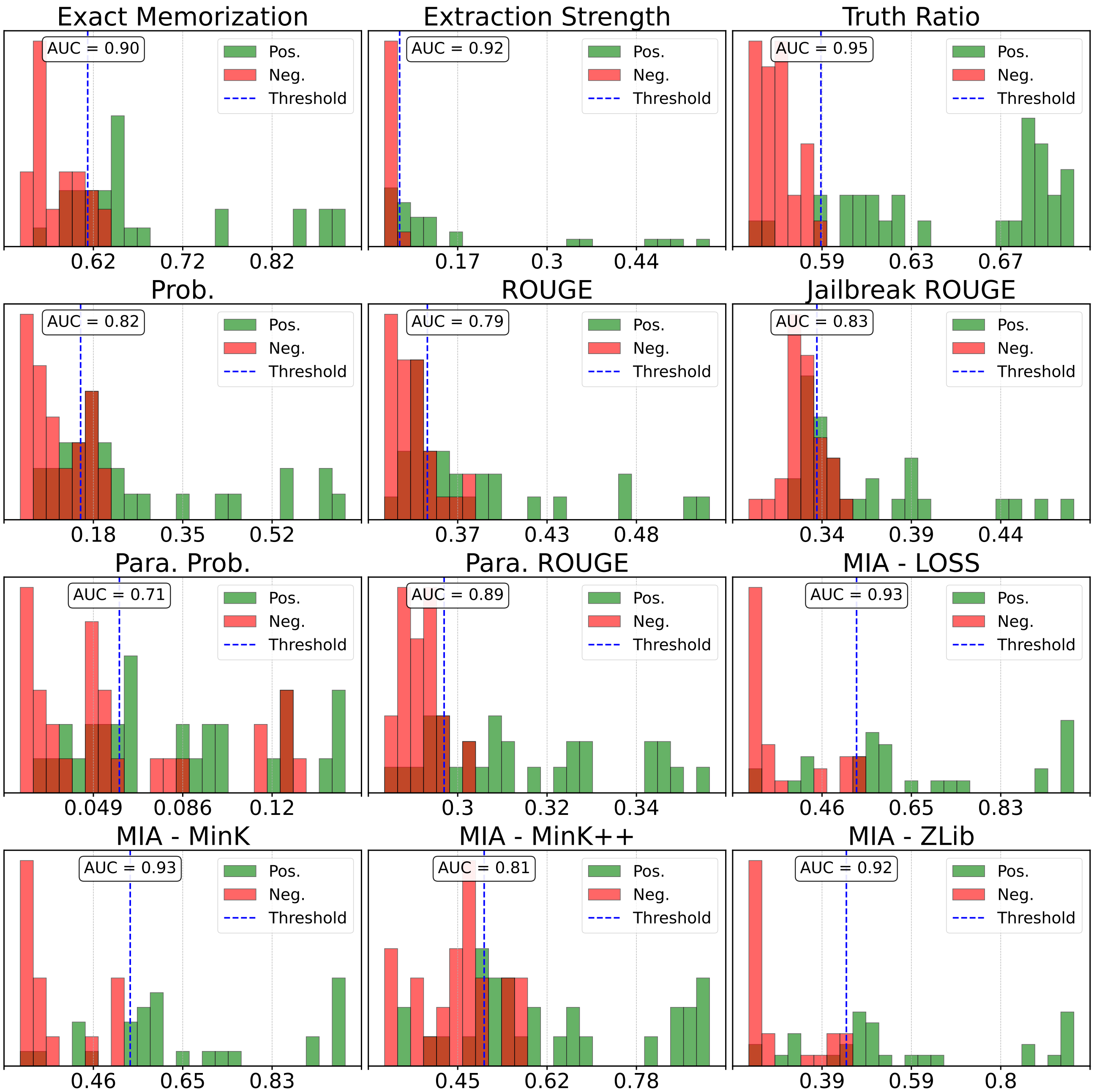

Figure 8. Faithfulness: Evaluation of multiple metrics to assess faithfulness. AUC indicates how effectively metrics distinguish between models trained on the target knowledge and those that are not.

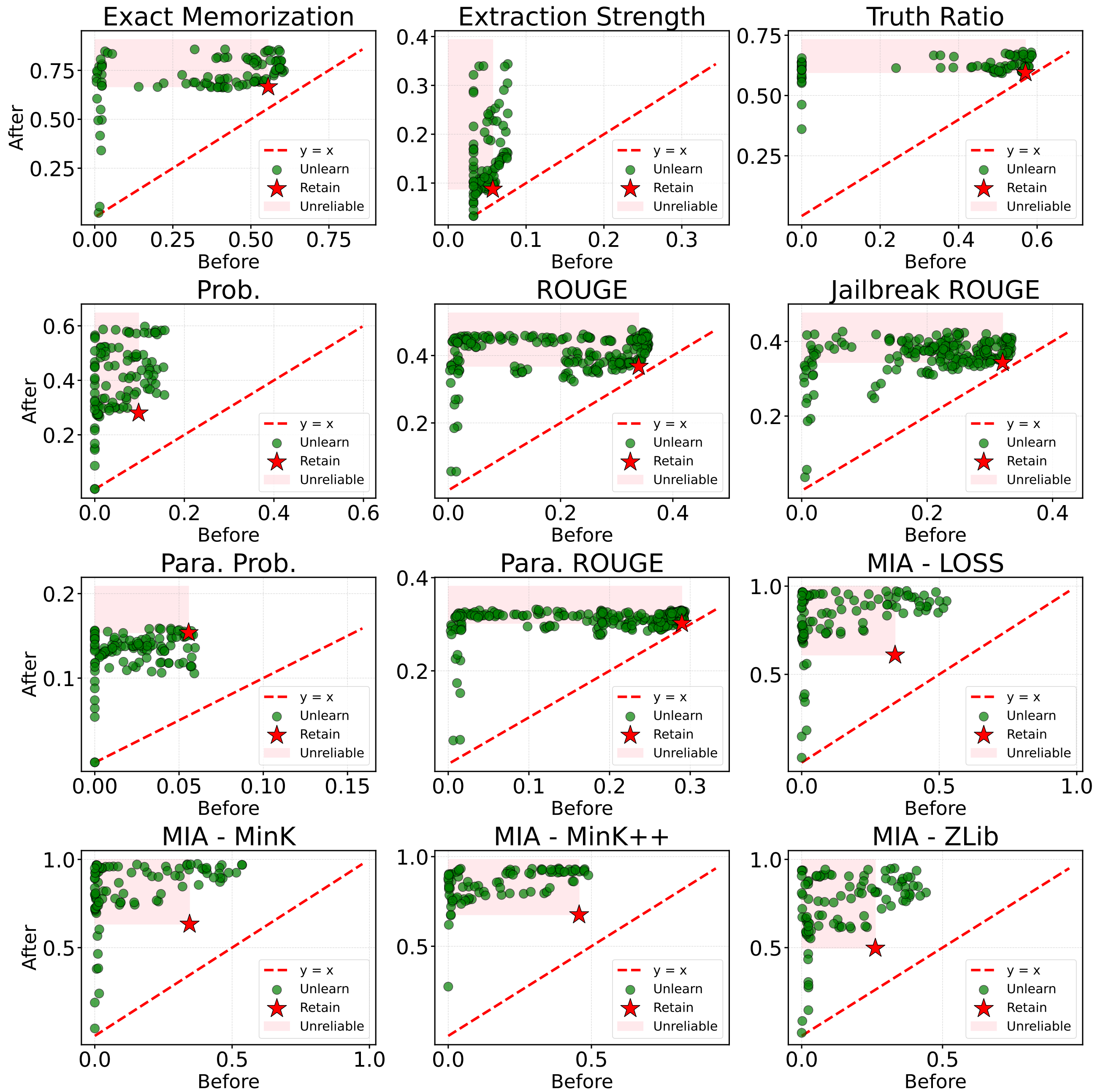

Figure 9. Relearning: Stress-testing multiple evaluation metrics through relearning. A significant fraction of unlearned models regain knowledge faster than the retained model when re-exposed to the forgotten data, falling into the unreliable red-shaded region: indicating that the metrics failed to initially capture the knowledge and are thus not robust.

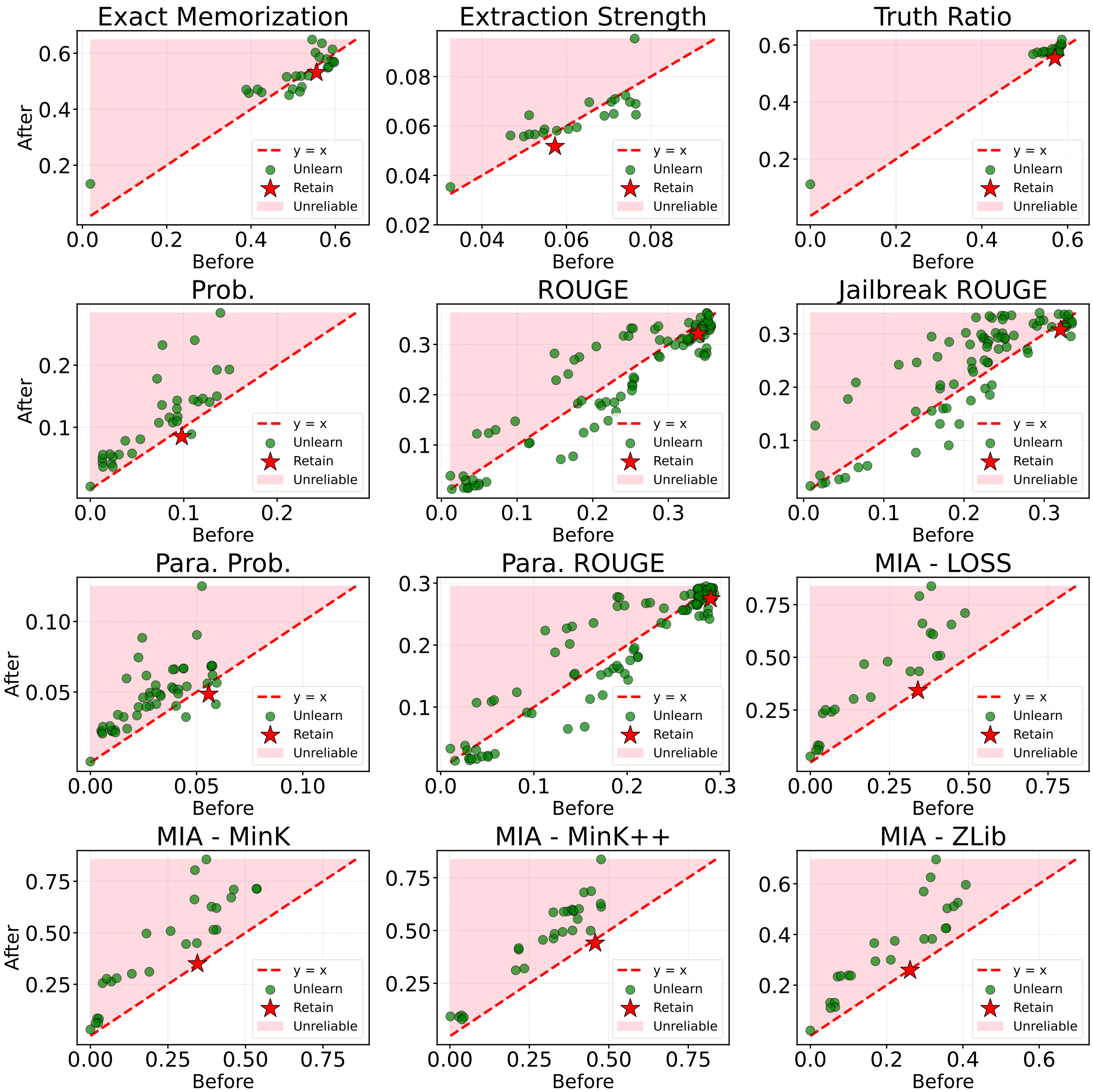

Figure 10. Quantization: Stress-testing multiple evaluation metrics through quantization. For several metrics, a subset of unlearned models shows increased metric values after quantization, falling into the red-shaded region: suggesting that the metrics failed to initially capture the presence of knowledge and are therefore not robust. These results are reported only for models unlearned with low learning rates and high utility.

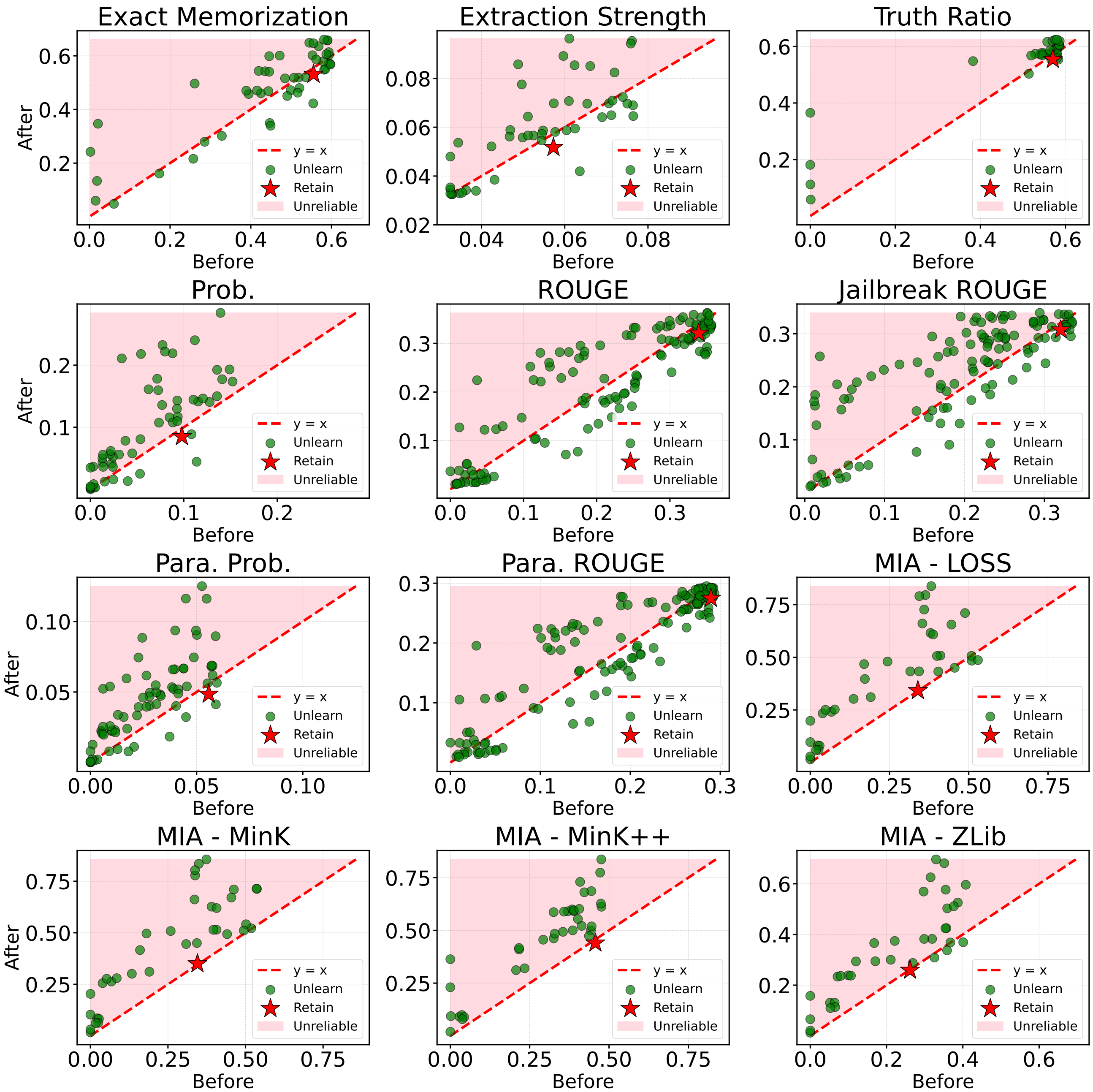

Figure 11. Quantization: Stress-testing multiple evaluation metrics through quantization. For each metric, a subset of unlearned models shows increased metric values after quantization, falling into the red-shaded region, suggesting that the metrics failed to initially capture the presence of knowledge and are therefore not robust. These results are reported only for models unlearned with low learning rates and no filter on utility.

G. Benchmarking unlearning methods

Unlike prior works with limited baselines and metrics, `OpenUnlearning` provides a standardized and scalable framework to conduct a large-scale comparison of various unlearning methods. We demonstrate this by evaluating 8 unlearning methods across 10 metrics on TOFU.

G.1. Unlearning methods

`OpenUnlearning` enables evaluation across a broader range of methods, including SimNPO (Fan et al., 2024), RMU (Li et al., 2024), AltPO (Mekala et al., 2025), NPO (Zhang et al., 2024), UNDIAL (Dong et al., 2025), as well as baselines like IdkPO, IdkNLL, and GradDiff (Maini et al., 2024). See Appendix D.7 for each method's definition.

G.2. Evaluation metrics

We evaluate unlearning methods using memorization metrics validated in our meta-analysis, alongside privacy and utility metrics. Using the TOFU benchmark and following the SemEval 2025 LLM Unlearning Challenge's ranking procedure (Ramakrishna et al., 2025b), we compute a composite score by aggregating metrics from the three categories:

1. **Memorization**: To quantify the degree of successful forgetting, the Memorization Score is calculated as the Harmonic Mean (HM) of 4 core metrics which are best as per our meta-evaluations analysis in § 1 — ES, EM, Paraphrased Probability and Truth Ratio. These metrics are inverted (i.e., $1 -$ metric) so that higher scores indicate more effective unlearning. The score is given by:

$$\text{Memorization Score} = \text{HM}\left(1 - \text{ES},\ 1 - \text{EM},\ 1 - \text{Para. Prob},\ 1 - \text{Truth Ratio}\right)$$

   Note that the memorization score (reported in Table 6) corresponds to forgetting: higher memorization indicates less knowledge.

2. **Privacy**: For assessing privacy, we utilize four Membership Inference Attack (MIA) metrics: LOSS, ZLib, Min-k, and Mink++. For each of these, an individual privacy score (s_{MIA}) is calculated. This score, ranging from 0 to 1, quantifies how closely the unlearned model's behavior on the specific MIA metric aligns with that of a gold-standard retain model (details below). A higher s_{MIA} score indicates greater similarity to the retain model. The overall Privacy Score is then the Harmonic Mean (HM) of these individual scores:

$$\text{Privacy Score} = \text{HM}(s_{\text{LOSS}}, s_{\text{ZLib}}, s_{\text{Min-k}}, s_{\text{Mink++}})$$

3. **Utility**: TOFU evaluates a model's utility using nine core metrics that assess performance across splits at three different distances from the forget dataset distribution - namely, retain, real-world authors, and wrong-fact queries: using QA probability, ROUGE, and truth-ratio scores. In addition to this we include a new metric that measures the fluency of the model's response when prompted with entities-related to forget queries, following (Mekala et al., 2025; Gandikota et al., 2024). Fluency is assessed using a classifier[5] that detects gibberish / nonsensical outputs. The final utility score is the harmonic mean of MU and fluency. Note that we scale all metrics with init finetuned model, so their scores across all points fall in the $[0, 1]$ range. For example, TOFU MU scores never exceed that of the initial target model upon unlearning, so all scores are effectively divided by the target model's MU.

Note that for many metric aggregations we use Harmonic Mean, as HM ensures that a high final score demands strong performance in all constituent parts.

G.3. Hyper-parameters tuning and model selection

Tuning: To ensure fairness, 27 hyperparameter tuning trials are allocated per method, as tuning can significantly improve performance of even simple baselines (Wang et al., 2025a).

1. For GradDiff and IdK-NLL: we vary the learning rate over the set $\{1\times10^{-5}, 2\times10^{-5}, 3\times10^{-5}, 4\times10^{-5}, 5\times10^{-5}\}$, and sweep the regularization coefficient $\alpha \in \{1, 2, 5, 10\}$.

[5] `https://huggingface.co/madhurjindal/autonlp-Gibberish-Detector-492513457`

Table 5. Comparison of unlearning methods on the TOFU task, showing aggregate (Agg.) using only Memorization (Mem.) and utility (Utility) scores. Privacy scores are not used in the aggregation and are only shown for illustration. Higher scores indicate better performance (↑). Initial finetuned is the target model before unlearning and Retain model is the gold standard target model. The focus on memorization as opposed to privacy results in GradDiff performing the best as it easily results in over-unlearning.

Method	Agg. ↑	Mem. ↑	Priv. ↑	Utility ↑
Init. finetuned	0.00	0.00	0.10	1.00
Retain	0.58	0.31	1.00	0.99
GradDiff (Maini et al., 2024)	**0.87**	**0.97**	3.27e-03	0.79
AltPO (Mekala et al., 2025)	<u>0.76</u>	<u>0.63</u>	0.06	<u>0.95</u>
IdkDPO (Maini et al., 2024)	0.71	0.56	0.06	<u>0.95</u>
NPO (Zhang et al., 2024)	0.69	0.52	0.06	**0.99**
RMU (Li et al., 2024)	0.53	0.47	0.5	0.61
SimNPO (Fan et al., 2024)	0.49	0.32	<u>0.63</u>	1.0
UNDIAL (Dong et al., 2025)	0.4	0.27	0.48	0.78
IdkNLL (Maini et al., 2024)	0.14	0.08	0.17	0.93

2. For IdK-DPO, NPO and AltPO: we tune learning rates in $\{1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}\}$, and search over $\alpha \in \{1, 2, 5\}$ and $\beta \in \{0.05, 0.1, 0.5\}$.
3. For RMU: we use the same learning rate range $\{1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}\}$, vary the steering coefficient in $\{1, 10, 100\}$, and apply the loss at one of the layers $l \in \{6, 11, 16\}$ of the LLama3.2-1B model. For each selected layer l, we restrict training to layers $l - 2$, $l - 1$, and l.
4. For SimNPO: we tune learning rates in $\{1 \times 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}\}$, and search over $\beta \in \{3.5, 4.5\}$, $\delta \in \{0, 1\}$ and $\delta \in \{0.125, 0.25\}$.
5. For UNDIAL: we tune learning rates in $\{1 \times 10^{-5}, 1 \times 10^{-4}, 3 \times 10^{-4}\}$, and search over $\alpha \in \{1, 2, 5\}$ and $\beta \in \{3, 10, 30\}$.

We aggregate utility score and memorization score and use their harmonic mean for tuning the models.

What metrics are appropriate for model selection during hyperparameter tuning? The nature of tuning in unlearning benchmarking has distinct considerations compared to general machine learning. While standard machine learning avoids using test data for tuning to ensure generalization, unlearning in TOFU and MUSE specifically targets the known forget set for erasure. Consequently, iteratively refining the unlearning by evaluating the model's behavior concerning this specific set is a permissible approach to ensure thorough forgetting before deployment. For this tuning, we advocate relying on metrics realistically available during the development phase, specifically those assessing forget quality on the target data and general utility, while avoiding "oracle" metrics that presume access to unavailable resources like true i.i.d holdout sets or retain models like in (Maini et al., 2024; Shi et al., 2025). Since all our privacy scores use a retain model, we avoid them during tuning. We rely on the harmonic mean of the Memorization and Utility scores as the validation objective.

Comparison to Wang et al. (2025a)'s benchmarking: While Wang et al. (2025a) propose approaches towards model selection and benchmarking through validation on Extraction Strength and calibration via model-merging, their analysis has several limitations. They rely only on ES scores for evaluating forgetting and utility. ES was found to be robust among the set of 4 evaluation metrics (an observation also re-verified in our work (§3). Yet it has not been proved that ES is a comprehensive metric validating all facets of knowledge unlearning. For example, ES does not account for privacy metrics that prevent over-unlearning, like TOFU's Truth Ratio or FQ or MUSE's PrivLeak. In addition, they do not consider all facets of general utility evaluation, particularly forget set fluency. Finally, the question of what metrics can be used in model selection and if they must be separate from the leaderboard metrics remains unanswered. These limitations remain, to a smaller degree, in our benchmarking procedure, and we consider this an important line for further research.

G.4. Results and discussion

While memorization, privacy, and utility each capture a distinct aspect of unlearning quality, we follow Ramakrishna et al. (2025b) in reporting their simple average as our aggregate score. On this composite metric (Table 6), SimNPO (Fan et al., 2024) ranks first. Although its memorization score trails that of others, it remains close to the retain model's level, showing

Table 6. Comparison of unlearning methods on the TOFU task, showing overall aggregate (Agg.), memorization (Mem.), privacy (Priv.), and utility (Utility) scores. Higher scores indicate better performance (↑). Initial finetuned is the target model before unlearning and Retain model is the gold standard target model. The best values are shown in bold, and the second-best values are underlined.

Method	**Agg.** ↑	**Mem.** ↑	**Priv.** ↑	**Utility** ↑
Init. finetuned	0.00	0.00	0.10	1.00
Retain	0.58	0.31	1.00	0.99
SimNPO (Fan et al., 2024)	**0.53**	0.32	**0.63**	**1.00**
RMU (Li et al., 2024)	<u>0.52</u>	0.47	<u>0.50</u>	0.61
UNDIAL (Dong et al., 2025)	0.42	0.27	0.48	0.78
AltPO (Mekala et al., 2025)	0.15	<u>0.63</u>	0.06	0.95
IdkNLL (Maini et al., 2024)	0.15	0.08	0.17	0.93
NPO (Zhang et al., 2024)	0.15	0.52	0.06	<u>0.99</u>
IdkDPO (Maini et al., 2024)	0.14	0.56	0.06	0.95
GradDiff (Maini et al., 2024)	9e-3	**0.97**	3e-3	0.79

a lack of over-unlearning. SimNPO fully preserves utility and achieves competitive privacy results, striking a balance across all three criteria. The next best performer is RMU, which demonstrates strong memorization and privacy but suffers a significant drop in utility.

Methods that under-unlearn (e.g. IdkNLL, which yields a low memorization score i.e. less forgetting) score lower on privacy. On the other hand, methods that over-unlearn (e.g. GradDiff, which forgets too strongly, reaching high memorization score) has poor privacy results as it deviates excessively from the retain model's behavior. This suggests that detecting and halting unlearning once the model's behavior has reverted to its "default" state is crucial to ensure privacy.

Because different ranking schemes can produce very different rankings (Table 6 v/s Table 7), it is critical to choose an appropriate method ranking procedure and aggregate metrics. Additionally, there is a lack of standardization on which metrics are suitable for model selection versus final evaluation (elaborated upon in Appendix G.3). While identifying the ideal ranking method and model selection approach is beyond our scope, we release all unlearned model checkpoints from our study to support future research on fair evaluation.

Table 7. Comparison of unlearning methods on the TOFU task, showing aggregate (Agg.) using only Memorization (Mem.) and utility (Utility) scores. Privacy scores are not used in the aggregation and are only shown for illustration. Higher scores indicate better performance (↑). Initial finetuned is the target model before unlearning and Retain model is the gold standard target model. The focus on memorization as opposed to privacy results in GradDiff performing the best as it easily results in over-unlearning.

Method	**Agg.** ↑	**Mem.** ↑	**Priv.** ↑	**Utility** ↑
Init. finetuned	0.00	0.00	0.10	1.00
Retain	0.58	0.31	1.00	0.99
GradDiff (Maini et al., 2024)	**0.87**	**0.97**	3.27e-03	0.79
AltPO (Mekala et al., 2025)	<u>0.76</u>	<u>0.63</u>	0.06	<u>0.95</u>
IdkDPO (Maini et al., 2024)	0.71	0.56	0.06	<u>0.95</u>
NPO (Zhang et al., 2024)	0.69	0.52	0.06	**0.99**
RMU (Li et al., 2024)	0.53	0.47	0.5	0.61
SimNPO (Fan et al., 2024)	0.49	0.32	<u>0.63</u>	1.0
UNDIAL (Dong et al., 2025)	0.4	0.27	0.48	0.78
IdkNLL (Maini et al., 2024)	0.14	0.08	0.17	0.93