Position: The Term "Machine Unlearning" Is Overused in LLMs

Anonymous Authors[1]

Abstract

This position paper argues that *machine unlearning* is overused as a term in LLM research and should be reserved for dataset-defined deletion: removing the training influence of a precisely specified forget set such that the resulting model is (approximately) indistinguishable from retraining without that data. We contend that many tasks currently labeled "unlearning" (e.g., refusal for harmful requests, entity/knowledge removal, or targeted suppression) pursue different, often policy-dependent objectives and therefore require different terminology and baselines (e.g., alignment, suppression, editing, obfuscation). We further argue that this confusion is not cosmetic: because papers make different implicit guarantees under the same label, metrics and benchmarks are frequently reused outside their intended scope, rewarding surface-level non-disclosure even when retraining-equivalence is not tested and derived capabilities remain. We conclude by calling for stricter terminology tied to explicit guarantees and reference models, and for evaluations that match the claimed objective.

1. Introduction

Foundation models are trained on large, heterogeneous corpora assembled under mixed licenses, consents, and contractual constraints. As these models are deployed in regulated and commercial settings, service providers increasingly face requests to *remove* the effect of specific training data, motivated by privacy deletion obligations (e.g., the right to be forgotten), copyright and licensing disputes (*Tremblay v. OpenAI, Inc.*, 2023), and enterprise data-governance requirements (Voigt & Von dem Bussche, 2017). These pressures have sharpened interest in *machine unlearning* as a principled way to remove the influence of selected training data.

In the classical machine learning formulation, *machine unlearning* is a dataset-defined deletion problem. Given a training set D and a precisely specified *forget set* $F \subset D$, the goal is to produce an updated model whose behavior is (approximately) indistinguishable from the counterfactual model obtained by retraining from scratch on $D \setminus F$ (Ginart et al., 2019). This definition fixes both the *target* and the *baseline*: it requires removing the training influence of a concrete subset of data, and it judges success by similarity to a model retrained on $D \setminus F$ (or a principled proxy), rather than by whether the outputs satisfy a chosen policy.

However, in recent LLM research, the word "unlearning" is frequently used for a broader range of objectives that share a high-level motivation ("make the model forget X") but do not match the retraining-based guarantee. Examples include preventing harmful behaviors, suppressing specific knowledge, removing entities, or blocking classes of queries (Li et al., 2024; Jin et al., 2024; Choi et al., 2025). These directions are important in practice, especially for safety and product policy, but they typically target *behavioral constraints* rather than dataset-defined deletion. When these objectives are discussed under the same term as machine unlearning, claims and evaluations become difficult to interpret: readers cannot tell whether a method aims to match retraining on $D \setminus F$, or merely to change what the system says under a particular prompting protocol.

A central reason is that many non-compliance "forgetting" requests are inherently *policy-defined* and application-dependent (Li et al., 2024; Jin et al., 2024; Luo et al., 2025). For instance, "forget harmful behavior" (e.g., bomb-making assistance) requires choosing a boundary: should the system block only step-by-step weaponization instructions, or also broadly relevant chemistry knowledge? Likewise, "forget knowledge" is ambiguous under entailment: if the target is "Paris is the capital of France," should the system also avoid entailed statements such as "the Eiffel Tower is in the capital of France"? Entity removal is similarly underspecified: "forget Stephen King" could refer to biographical facts, his works, quotations, or derivative discussion. This subjectivity can make it difficult to specify a precise forget set, but that is not the core issue. More fundamentally, the objective is defined by an application policy (i.e., what the model should or should not do) so the problem is inherently about policy compliance rather than dataset-defined deletion.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

The gap between dataset-defined deletion and policy-defined behavior control is clearest for *derived capabilities*, where training influence is not limited to memorizing the forget set. For example, suppose a model is trained on unauthorized synthetic mathematical reasoning traces and is later required to "unlearn" them. If evaluation only checks whether the model fails to answer the same questions from that dataset, a trivial non-disclosure strategy can appear successful. The relevant question is whether the unauthorized data contributed to a *transferable* reasoning capability: if the model still solves broad classes of challenging math problems, influence may persist even when direct reproduction is blocked. Under retraining-indistinguishability, maintaining such a capability is acceptable only if the retrained model on $D \setminus F$ achieves it; otherwise, the capability should disappear along with the influence that induced it.

This terminological ambiguity also directly affects benchmarks and metrics. Many evaluations operationalize "forgetting" as output failure on a designated probe set (Jin et al., 2024; Yuan et al., 2025; Xu et al., 2026). Such measures can be useful diagnostics for non-disclosure, but they are not evidence of retraining equivalence. They are also often subjective, and *lower is not always better*: a retrained model on $D \setminus F$ may still produce partially correct or contextually reasonable answers, while a blanket refusal can drive ROUGE toward zero while diverging from the retrained reference. Benchmarks therefore add retain/utility constraints (Maini et al., 2024; Shi et al., 2025; Chang & Lee, 2025), but these too encode application-dependent choices about what counts as utility and what trade-offs are acceptable. Without an explicit retrain reference, evaluation can unintentionally prioritize output control over removal of training influence.

In this position paper, we argue that resolving this confusion requires stricter terminology tied to explicit guarantees and baselines. We formalize machine unlearning as retraining-indistinguishability for a precisely defined forget set, organize other common "unlearning" usages by intent, and explain why benchmark design must reflect the distinction, especially in the presence of derived capabilities.

Position. "Machine unlearning" should mean retraining indistinguishability for a precisely defined forget set; other safety- or application-driven "forgetting" goals are different problems and should use different terms.

2. What the Literature Calls "Unlearning": A Definition and a Taxonomy by Intent

In this section, we (i) give a *formal definition* of **machine unlearning**, and (ii) organize other common uses of the term into *high-level categories by intent*, without attempting rigid, mutually exclusive formalization.

2.1. Unlearning: Dataset-Defined Deletion Guarantee

Setup. Let D be the training dataset and let $F \subseteq D$ be the *forget set*, whose *training influence* is to be removed. Define the retain set as $R := D \setminus F$. Let $\mathsf{Train}(\cdot)$ denote the (randomized) training procedure, and write $\Theta_S \sim \mathsf{Train}(S)$ for the random model obtained by training on dataset S. An unlearning is a (possibly randomized) procedure that takes a trained model and a forget set and returns an updated model:

$$\Theta' \leftarrow \mathsf{Unlearn}(\Theta_D, F).$$

Informally, machine unlearning aims to remove the influence of training on F as if the model had never seen it.

Definition 2.1 (Exact machine unlearning (Izzo et al., 2021))**.** Unlearn achieves *exact machine unlearning* (with respect to Train) if for all D and all $F \subseteq D$,

$$\mathcal{L}(\Theta') = \mathcal{L}(\Theta_R), \quad \text{where } \Theta_R \sim \mathsf{Train}(R),$$

and $\mathcal{L}(\cdot)$ denotes the induced distribution over model parameters (and over the randomized training outcome).

In practice, exact unlearning is a very strong requirement and is rarely attainable for large-scale models. Accordingly, most work adopts relaxed notions of unlearning that allow the unlearned model to be *approximately* indistinguishable from the retrained baseline.

Definition 2.2 (Approximate machine unlearning (general form))**.** Fix a divergence/metric Dist between distributions and a tolerance $\tau \geq 0$. Unlearn achieves (Dist, τ)-*approximate machine unlearning* if for all D and $F \subseteq D$,

$$\mathrm{Dist}(\mathcal{L}(\Theta'),\, \mathcal{L}(\Theta_R)) \leq \tau.$$

Dist may be defined in parameter space or in *behavior space*. We emphasize that *multiple* choices of Dist are reasonable; the key is that the baseline is always retraining on $D \setminus F$.

One widely used relaxation (inspired by differential privacy (Dwork, 2006)) defines closeness via (ε, δ)-indistinguishability. For random variables X and Y, write $X \approx_{\varepsilon,\delta} Y$ if for all measurable sets S,

$$\Pr[X \in S] \leq e^{\varepsilon} \Pr[Y \in S] + \delta$$
$$\Pr[Y \in S] \leq e^{\varepsilon} \Pr[X \in S] + \delta.$$

Then Unlearn is (ε, δ)-*approximate* if $\Theta' \approx_{\varepsilon,\delta} \Theta_R$. This is a principled and popular choice, but it is *not the only* way to formalize approximate unlearning.

2.2. Other Common Uses of "Unlearning" in LLM Papers: Categories by Intent

We now summarize several common intents that are frequently labeled "unlearning" in the LLM literature.

Output Likelihood Suppression. Suppression methods directly lower the likelihood of forget-related outputs, typically via gradient ascent (Jang et al., 2023) or negative preference optimization (Zhang et al., 2024). They can shift probability mass away from restricted responses, but mainly operate on output distributions.

Internal Representation Obfuscation. Obfuscation methods make forget-related inputs unreliable by inducing distorted activations (Li et al., 2024; Zou et al., 2024; Wuerkaixi et al., 2025) or high-entropy predictions (Yuan et al., 2025; Entesari et al., 2025). They reduce answerability, but do not necessarily recover retrained-model behavior.

Knowledge Editing. Editing methods modify semantic associations through knowledge editing (Li et al., 2025; Hossain & Kagal, 2025; Jung et al., 2025) or replacement supervision such as counterfactual fine-tuning (Eldan & Russinovich, 2023; Gu et al., 2024; Scholten et al., 2025).

Behavioral Refusal. Refusal methods train the model to abstain on forget-related queries, often with "I don't know" responses (Maini et al., 2024; Yuan et al., 2025). They mainly change policy rather than learned content.

3. Why Terminology Matters for LLM Unlearning Evaluation

When suppression, refusal, editing, and machine unlearning are all grouped under "unlearning," benchmarks often reduce success to what is easiest to measure: whether the model fails to produce a reference answer on forget queries.

3.1. Output-Failure Metrics Dominate Current Practice

Most LLM "unlearning" evaluations measure success by output failure, using metrics such as surface similarity, embedding similarity, or likelihood of the reference answer. While these scores capture what the model emits under fixed prompts, they do not establish retrain-equivalent unlearning without comparison to a retrained model. As a result, suppression, refusal, or editing methods can be rewarded as "better unlearning" even when they only reduce benchmark-specific output reproduction.

3.2. What Leading Benchmarks Actually Measure

TOFU (Maini et al., 2024) is one of the few benchmarks that includes a retrained reference. Its forget quality metric compares unlearned and retrained models using probability-based truth ratios and distributional tests. However, later works often use TOFU without the retrained baseline (Yuan et al., 2025), turning the evaluation into output non-reproduction rather than retrain-equivalence.

MUSE (Shi et al., 2025) evaluates multiple criteria, including verbatim memorization, knowledge memorization, privacy leakage, and utility. It includes a retrained baseline for membership-inference-based privacy leakage, but its memorization metrics rely largely on ROUGE reduction on forget queries. Parts of the benchmark measure retrain-equivalence, while others primarily measure output suppression.

RWKU (Jin et al., 2024) evaluates real-world knowledge removal using QA, cloze prompts, MIAs, and adversarial elicitation. Since it does not use a retrained reference, its scores measure robustness of knowledge suppression under diverse probes rather than the retrain-equivalence.

WMDP (Li et al., 2024) frames unlearning as reducing hazardous capabilities, measured by lower QA accuracy in biosecurity, cybersecurity, and related domains. Because it does not correspond to deletion of a defined training subset, WMDP is better understood as capability suppression.

3.3. Adversarial Evaluation Exposes the Limitations of Output-Failure Scores

The gap between output failure and influence removal becomes clear under stress testing. Methods that appear successful under fixed prompts often fail under paraphrasing, mixed queries, or jailbreak-style elicitation (Lynch et al., 2024; Łucki et al., 2025; Jeung et al., 2025), suggesting that the target knowledge was suppressed rather than removed.

Model-level interventions reveal the same issue. Additional fine-tuning (Yoon et al., 2026), post-training transformations such as quantization (Zhang et al., 2025), activation-level extraction (Seyitoğlu et al., 2024), and representation-level auditing (Goel et al., 2026) can recover information that output-failure metrics treated as forgotten.

4. Derived Capabilities: Unlearning Beyond Surface-Level Outputs

4.1. What We Mean by "Derived Capabilities"

We use the term *derived capability* to denote a form of behavioral competence that is plausibly attributable to training on F (or its interaction with the rest of training), and that generalizes beyond the exact examples contained in F. Such capabilities need not take the form of verbatim memorization. They may instead manifest as transferable reasoning skills learned from reasoning traces, trigger-conditioned or adversarial behaviors induced by a small number of poisoned samples, persistent stylistic or tool-use habits, or latent factual competence that can be recovered under paraphrase or benign post-training interventions even when direct regurgitation is suppressed. These phenomena are conceptually important because they reveal why "unlearning = not answering" is an incomplete operationalization.

4.2. Implications of the Machine-Unlearning Definition

Machine unlearning is defined by approximate equivalence to retraining on $R = D \setminus F$, which entails the following implication: *If training on F contributes to a derived capability, then an unlearned model that is truly indistinguishable from retraining without F must also lose that capability.* This can be undesirable when F is entangled with useful behavior, but it follows directly from retraining counterfactual: the reference model is one that never learned from F. This marks a key boundary. Methods that preserve capabilities attributable to F may still be useful for suppression, editing, or alignment, but they do not satisfy machine unlearning.

4.3. Case: Unauthorized Synthetic Reasoning Traces

Derived capabilities are especially salient when training involves synthetic supervision. Consider a setting in which mathematical reasoning traces generated by a frontier LLM are used, without authorization, to train another model, resulting in a measurable improvement in mathematical reasoning performance. In practice, several model providers explicitly prohibit the use of their outputs to train or fine-tune competing models, and such use may later trigger a request to "unlearn" the unauthorized data.

In this scenario, the central question is not whether the model can reproduce specific solutions from the synthetic dataset. Rather, it is whether the unauthorized reasoning traces contributed to a general mathematical reasoning capability that transfers beyond the original examples.

If evaluation treats output failure on the unauthorized data as the sole success criterion, a model can appear successfully unlearned by simply refusing to answer or producing irrelevant responses, while retaining the improved reasoning ability. Such evaluation therefore fails to test whether the training influence on a transferable capability has been removed. This case illustrates why output non-reproduction is an unreliable proxy for unlearning whenever the effect of the forget set manifests as a derived capability.

5. Call for Action: Reference-Based Evaluation and Derived-Capability Probes

5.1. Evaluate Unlearning Against a Reference Model

Success should be judged by similarity to a reference model approximating the model trained on $D \setminus F$. Ideally, this is a model retrained from scratch on $D \setminus F$; when infeasible, papers should use the strongest proxy and state it explicitly. Output-level metrics are not flawed, but they should not be used as a stand-alone unlearning criterion: they mainly measure output control under the probe distribution rather than removal of training influence. Papers claiming machine unlearning, rather than output control, should report:

A reference model, with provenance: ideally $\mathsf{Train}(D \setminus F)$ with matched hyperparameters. If this is infeasible, use the best available proxy (e.g., a matched-stage retrain for fine-tuning unlearning, a smaller-scale retraining study, or the strongest checkpoint *before* the introduction of F) and clearly state what it approximates and what it does not.

Distances to the reference: in addition to forget-query scores, report distributional comparisons to the reference model (e.g., logit- or probability-based statistics, MIA-style audits, and robustness under adversarial elicitation), since the question is similarity to the counterfactual baseline.

Utility relative to the reference: utility should be reported as part of the comparison to the reference model (not merely as a separate "do not break the model" constraint). Reference models are also needed to interpret capability-level effects: if a capability is absent from the counterfactual reference, retaining it after "unlearning" indicates residual influence; if it is present, unlearning should preserve it.

5.2. Derived-Capability Probes Should Be First-Class

When the claim is removal of training influence, evaluation should probe derived capabilities. Output failure is insufficient whenever F induces transferable behavior. We therefore recommend three complementary probes: (i) *capability-level holdout tasks*, which test capabilities plausibly induced by F on prompts disjoint from the forget set, such as out-of-distribution math problems in the synthetic reasoning case; (ii) *intervention-based recovery tests*, which check whether supposedly forgotten information or behaviors can be recovered by benign fine-tuning or post-training transformations; and (iii) *task-appropriate threat models*, which directly measure induced behaviors such as trigger success, targeted error, or poison influence in poisoning and backdoor settings. As with all metrics, they should be interpreted relative to the retrained reference, since the goal is to match the counterfactual behavior of training on $D \setminus F$.

6. Conclusion

We argue that *machine unlearning* should refer specifically to dataset-defined deletion: given a forget set $F \subset D$, the goal is to remove its training influence by matching the counterfactual model retrained on $D \setminus F$. Other forgetting objectives, such as suppression, editing, refusal, or filtering, may be useful but require distinct terminology and guarantees. This distinction matters because many benchmarks reward surface-level non-disclosure, which can miss persistent influence and derived capabilities. Accordingly, unlearning claims should be evaluated against an explicit retraining reference or stated proxy, with derived-capability probes when influence removal is the intended goal.

References

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

Choi, M., Rim, D., Lee, D., and Choo, J. Opt-out: Investigating entity-level unlearning for large language models via optimal transport. In *ACL main*, 2025.

Dwork, C. Differential privacy. In *ICALP*, 2006.

Eldan, R. and Russinovich, M. Who's harry potter? approximate unlearning in llms, 2023.

Tremblay v. OpenAI, Inc. 23-cv-03416-AMO, (N.D. Cal.), 2023.

Entesari, T., Hatami, A., Khaziev, R., Ramakrishna, A., and Fazlyab, M. Constrained entropic unlearning: A primal-dual framework for large language models. In *NeurIPS*, 2025.

Ginart, A., Guan, M., Valiant, G., and Zou, J. Y. Making ai forget you: Data deletion in machine learning. In *NeurIPS*, 2019.

Goel, A., Ritter, A., and Gurevych, I. Auditing language model unlearning via information decomposition. *arXiv preprint arXiv:2601.15111*, 2026.

Gu, T., Huang, K., Luo, R., Yao, Y., Yang, Y., Teng, Y., and Wang, Y. Meow: Memory supervised llm unlearning via inverted facts. *arXiv preprint arXiv:2409.11844*, 2024.

Hossain, S. and Kagal, L. Investigating model editing for unlearning in large language models. In *COLM Workshop on SoLaR*, 2025.

Izzo, Z., Smart, M. A., Chaudhuri, K., and Zou, J. Approximate data deletion from machine learning models. In *AISTATS*, 2021.

Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., and Seo, M. Knowledge unlearning for mitigating privacy risks in language models. In *ACL*, 2023.

Jeung, W., Yoon, S., and No, A. Seps: A separability measure for robust unlearning in llms. In *EMNLP main*, 2025.

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. In *NeurIPS*, 2024.

Jung, D., Seo, J., Lee, J., Park, C., and Lim, H. CoME: An unlearning-based approach to conflict-free model editing. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *NAACL main*, 2025.

Li, N., Pan, A., Gopal, A., Yue, S., Berrios, D., Gatti, A., Li, J. D., Dombrowski, A.-K., Goel, S., Mukobi, G., Helm-Burger, N., Lababidi, R., Justen, L., Liu, A. B., Chen, M., Barrass, I., Zhang, O., Zhu, X., Tamirisa, R., Bharathi, B., Herbert-Voss, A., Breuer, C. B., Zou, A., Mazeika, M., Wang, Z., Oswal, P., Lin, W., Hunt, A. A., Tienken-Harder, J., Shih, K. Y., Talley, K., Guan, J., Steneker, I., Campbell, D., Jokubaitis, B., Basart, S., Fitz, S., Kumaraguru, P., Karmakar, K. K., Tupakula, U., Varadharajan, V., Shoshitaishvili, Y., Ba, J., Esvelt, K. M., Wang, A., and Hendrycks, D. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *ICML*, 2024.

Li, Z., Wang, X., Shen, W. F., Kurmanji, M., Qiu, X., Cai, D., Wu, C., and Lane, N. D. Editing as unlearning: Are knowledge editing methods strong baselines for large language model unlearning? In *NeurIPS LLM Evaluation Workshop*, 2025.

Łucki, J., Wei, B., Huang, Y., Henderson, P., Tramèr, F., and Rando, J. An adversarial perspective on machine unlearning for ai safety. *TMLR*, 2025.

Luo, Y., Zhou, Z., Chen, H., Qiu, K., Savvides, M., Li, S., and Wang, J. Knowledgesmith: Uncovering knowledge updating in llms with model editing and unlearning. *arXiv preprint arXiv:2510.02392*, 2025.

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. In *COLM*, 2024.

Scholten, Y., Xhonneux, S., Schwinn, L., and Günnemann, S. Model collapse is not a bug but a feature in machine unlearning for llms. *arXiv preprint arXiv:2507.04219*, 2025.

Seyitoğlu, A., Kuvshinov, A., Schwinn, L., and Günnemann, S. Extracting unlearned information from LLMs with activation steering. In *NeurIPS Workshop SafeGenAi*, 2024.

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 *ICLR*, 2025.

Voigt, P. and Von dem Bussche, A. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer Publishing Company, Incorporated, 2017.

Wuerkaixi, A., Wang, Q., Cui, S., Xu, W., Han, B., Niu, G., Sugiyama, M., and Zhang, C. Adaptive localization of knowledge negation for continual llm unlearning. In *ICML*, 2025.

Xu, X., Du, M., Li, Z., Liang, Z., Guo, Z., Zhang, S., Hu, P., Ye, Q., and Hu, H. From domains to instances: Dual-granularity data synthesis for llm unlearning. *arXiv preprint arXiv:2601.04278*, 2026.

Yoon, S., Hong, H., Jeung, W., and No, A. Rethinking benign relearning: Syntax as the hidden driver of unlearning failures. In *ICLR*, 2026.

Yuan, X., Pang, T., Du, C., Chen, K., Zhang, W., and Lin, M. A closer look at machine unlearning for large language models. In *ICLR*, 2025.

Zhang, R., Lin, L., Bai, Y., and Mei, S. Negative preference optimization: From catastrophic collapse to effective unlearning. In *COLM*, 2024.

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 *ICLR*, 2025.

Zou, A., Phan, L., Wang, J., Duenas, D., Lin, M., Andriushchenko, M., Kolter, J. Z., Fredrikson, M., and Hendrycks, D. Improving alignment and robustness with circuit breakers. In *NeurIPS*, 2024.