The Erasure Illusion: Stress-Testing the Generalization of LLM Forgetting Evaluation

Hengrui Jia†, Taoran Li*, Jonas Guan†, Varun Chandrasekaran*
†*University of Toronto and Vector Institute, * University of Illinois Urbana-Champaign*

Abstract

Machine unlearning aims to remove specific data influences from trained models, a capability essential for adhering to copyright laws and ensuring AI safety. Current unlearning metrics typically measure success by monitoring the model's performance degradation on the specific unlearning dataset (D_u). We argue that for Large Language Models (LLMs), this evaluation paradigm is insufficient and potentially misleading. Many real-world uses of unlearning–motivated by copyright or safety–implicitly target not only verbatim content in D_u, but also behaviors influenced by the broader generalizations the model derived from it. We demonstrate that LLMs can pass standard unlearning evaluation and appear to have "forgotten" the target knowledge, while simultaneously retaining strong capabilities on content that is semantically adjacent to D_u. This phenomenon indicates that erasing exact sentences does not necessarily equate to removing the underlying knowledge. To address this gap, we propose Proximal Surrogate Generation (PSG), an automated stress-testing framework that generates a surrogate dataset, $\tilde{D}_u$. This surrogate set is constructed to be semantically derived from D_u yet sufficiently distinct in embedding space. By comparing unlearning metric scores between D_u and $\tilde{D}_u$, we can stress-test the reliability of the metric itself. Our extensive evaluation across three LLM families (Llama-3-8B, Qwen2.5-7B, and Zephyr-7B-β), three distinct datasets, and seven standard metrics reveals widespread inconsistencies. We find that current metrics frequently overestimate unlearning success, failing to detect retained knowledge exposed by our stress-test datasets.

1 Introduction

The rapidly increasing adoption of large language models (LLMs) in society raise severe privacy, compliance, and intellectual property concerns [8, 26, 35, 50]. Machine unlearning aims to address these concerns by selectively removing the influence of specific data points from a trained model such that it behaves as if it had never seen that data [4, 5]. This is particularly important in domains requiring post-hoc data deletion; for instance, to satisfy regulatory frameworks like the GDPR's "right to be forgotten" [45].

In many applications—particularly those motivated by copyright or safety—the data targeted for unlearning is often not merely surface-level content from an unlearning set D_u, but the higher-order knowledge inferred from D_u [30]. Beyond suppressing verbatim recitation, many real-world deployments implicitly expect unlearning to eliminate abstractions that allow the model to reconstruct or advantageously leverage information from D_u through paraphrase, composition, or transfer to semantically adjacent prompts [8]. Regulatory or ethical requirements may demand not only the removal of verbatim training examples, but also the elimination of learned behaviors such as authorial style, domain-specific reasoning patterns, or unsafe content generation [11].

Critically, LLM unlearning success is often evaluated only on the unlearning set D_u [36, 39, 39, 41, 43, 53], or small handcrafted proxies closely resembling it [11, 28, 33]. This risks yielding metric scores that only track surface recitation rather than the intended removal of higher-order knowledge (§ 3.1). This mismatch creates a false sense of unlearning: a model can satisfy these metrics while retaining capabilities that transfer to content semantically related to D_u (§ 6).

To better understand this gap, we propose the notion of a *surrogate unlearning dataset* $\tilde{D}_u$, which is disjoint but semantically-related to D_u. This allows us to examine whether existing metrics removed not only surface-level traces of D_u but also changes in related behaviors that practitioners often expect unlearning to influence. For example, many applications implicitly expect that unlearning a novel would also cause the model to perform consistently worse on fan fiction derived from that novel. We find that this is not the case: current unlearning metrics may judge the unlearning of a novel as successful, yet simultaneously report no notable unlearning on a derived fan fiction, despite their significant semantic overlap (§ 3.2).

Motivated by these insights, we propose Proximal Surrogate Generation (PSG) to automatically construct such surro-

gate datasets $\tilde{D}_u$ for any given unlearning set D_u and model θ (§ 4). Our approach simulates a lightweight fine-tuning process using D_u, and then samples text that the model becomes more confident in (i.e., shows increased likelihood). These samples are then perturbed, using techniques inspired by gradient-based control generation (e.g., GCG [55]) to increase their (embedding) distance from D_u. This creates semantically derivative but non-overlapping data. The resulting $\tilde{D}_u$ can be used to stress-test unlearning metrics: if a metric fails to behave consistently across D_u and $\tilde{D}_u$, it likely cannot reliably measure the success of unlearning.

We validate our findings across combinations of a range of LLMs, including Llama-3-8B [15], Qwen2.5-7B [51], and Zephyr-7B-beta [44], and 3 datasets to simulate different unlearning scenarios: a fantasy novel dataset, a toxic comment dataset [3], and a biomedical weapon dataset [28]. For each pair of model to be unlearned and D_u, we craft a $\tilde{D}_u$, consisting of sentences that can be learned from D_u, have sufficiently low perplexity, and are distant from sentences in D_u in the embedding space. After performing unlearning using Negative Preference Optimization (NPO) [54] and Representation Misdirection for Unlearning (RMU) [28], we evaluate the unlearned models on both D_u and $\tilde{D}_u$, using 7 common metrics for LLM unlearning. We observe that in 61.1% of the scenarios, the unlearning metrics are not consistent on D_u and $\tilde{D}_u$ (i.e., we define inconsistency as the two metric distributions' means differing by more than $1/2$ pooled standard deviation; § 6). In particular, the metric scores computed on $\tilde{D}_u$ are mostly higher than D_u, meaning they suggest $\tilde{D}_u$ is less successfully unlearned than D_u, despite $\tilde{D}_u$ being learned from D_u. Our code is available on github.

Our results suggest that the existence of meaningful surrogate datasets $\tilde{D}_u$ is not only plausible but also widespread. We do not propose a new definition of unlearning; rather, our results show that many practical expectations of LLM unlearning are not reflected in existing metrics. Thus, we advocate for a more nuanced understanding of what it means to unlearn in generative models, urging the community to distinguish between privacy-oriented deletion and knowledge-based unlearning. In doing so, we hope to inspire the design of new metrics that capture both direct and generalized influence of training data.

To summarize, our main contributions are as follows:

- We highlight a critical concern in LLM unlearning: current metrics that determine unlearning success are unreliable (§ 3.1); metrics can certify success on D_u yet report no notable unlearning on semantically-related set $\tilde{D}_u$ (§ 6).
- We propose a lightweight algorithm to construct a surrogate dataset $\tilde{D}_u$ for any unlearning set D_u and model to be unlearned θ, which can be used to audit the reliability of an unlearning metric on the dataset (§ 4).
- We empirically evaluate 7 common unlearning metrics on all combinations of 3 LLMs, 3 datasets, and 2 standard unlearning algorithms (NPO and RMU) and find that none of

the metrics consistently correlate with unlearning success; this shows the widespread impact of the metric mirage in LLM unlearning (§ 5 and 6).

2 Related Work

In this section, we provide the necessary background information for this work. We define the unlearning problem and introduce notation. We also list and briefly describe some frequently used unlearning methods and evaluation metrics that will be studied.

2.1 Background & Notation

What is the Unlearning Problem (for LLMs)? It is broadly defined as the mechanism to erase the "impact" of certain data points from a model (particularly, its behaviors). Depending on the incentives of unlearning (*e.g.,* privacy, copyright, etc.), this could either mean (a) **Data-level:** obtaining a model as if it were not trained on the exact data points of the unlearning set [2, 9, 19] (similar to guarantees provided by differential privacy [17, 18]) when the goal is to address privacy concerns, or (b) **Knowledge-level:** erasing all knowledge and behaviors that could be learned or derived from the unlearning set post-hoc when the goal is to *e.g.,* avoid copyright infringement or mitigate unsafe behaviors [24, 25, 29, 38, 46–48, 52]. In this work, we focus on the latter, as it is a less well-defined form of unlearning. Consequently, many existing evaluation methods for unlearning are designed for the former yet used to judge the latter, since LLMs generalize and the influence of the unlearning dataset appears beyond memorized points. Besides, proper evaluation methods for the former have already been studied and shown to be difficult [18].

As suggested in prior work [4], unlearning can trivially be achieved by retraining the model without the data of interest; such an approach constitutes "exact" unlearning. However, since this is computationally expensive, many "approximate/inexact" (and cheaper) alternatives exist. Concrete strategies will be discussed in § 2.2. It is also worth noting that "the data of interest" may not always be traceable for knowledge-level unlearning, because the given unlearning set often serves only as a "partial" proxy for the full capabilities stakeholders wish to remove; similar information may exist elsewhere in the rest of the training dataset.

Notation: Having established the problem, we now present the notation we use throughout the paper. Let $\mathcal{D}$ be the data distribution, $D \sim \mathcal{D}^m$ denote the dataset (of size m) used to train the model, $D_u \subseteq D$ be the unlearning dataset (i.e., whose contents we want to unlearn). and D_r be the retain dataset (i.e., whose information we want to retain; typically $D_r \subset D - D_u$). One can think of $D_u = \{z_1, \cdots, z_n\}$ where each $z_i = (x_i, y_i)$ where x_i is some prefix and y_i is some suffix. Let θ denote (the parameters of) the original (base) model trained on the full

dataset D. Then, the exact unlearned model (*e.g.,* obtained by training solely on $D - D_u$) is denoted by θ_u, and the approximately unlearned model is denoted by $\hat{\theta}_u$. A good approximate unlearning algorithm $\mathcal{U}$ aims to find $\hat{\theta}_u$ that is "close" to θ_u in terms of behavior and performance, particularly with respect to D_u and D_r. It is formally denoted by

$$\hat{\theta}_u = \mathcal{U}(\theta, D_u)$$

The effectiveness of $\mathcal{U}$ is measured by how well $\hat{\theta}_u$ satisfies unlearning desiderata, such as:

1. *Knowledge Removal*: The model $\hat{\theta}_u$ should demonstrate no discernible knowledge of D_u. For instance, its performance *e.g.,* perplexity (denoted $\mathcal{L}_p$) on D_u should be similar to that of a model not trained on D_u:

$$\mathbb{E}_{(x,y)\sim D_u}[\mathcal{L}_p(\hat{\theta}_u(x), y)] \approx \mathbb{E}_{(x,y)\sim D_u}[\mathcal{L}_p(\theta_u(x), y)]$$

2. *Utility Preservation*: The model $\hat{\theta}_u$ should maintain its performance on the retain set D_r:

$$\mathbb{E}_{(x,y)\sim D_r}[\mathcal{L}_p(\hat{\theta}_u(x), y)] \approx \mathbb{E}_{(x,y)\sim D_r}[\mathcal{L}_p(\theta(x), y)]$$

3. *Efficiency*: The cost of obtaining $\hat{\theta}_u$ from θ should be significantly less than retraining θ_u from scratch.

2.2 Unlearning Methods

We now provide a brief description of commonly-used approximate unlearning techniques for LLMs. Broadly speaking, these techniques often first identify the model parameters related to D_u, and then optimize these parameters to remove D_u's impacts by *e.g.,* gradient-based methods.

For instance, early approaches applied gradient ascent (GA) on D_u to reduce the model's log-probability of those examples [23]; the neurons with stronger gradient signal are naturally the ones related to D_u. While straightforward, this often causes severe degradation in generation quality. Huang *et al.* [20] propose gradient-based unlearning that decomposes the unlearning update into weighted gradient ascent on D_u and gradient descent on D_r to minimize utility degradation. Negative Preference Optimization (NPO) [54], a variant of preference optimization [37] that only uses negative samples, minimizes the learned reward for "undesired" outputs, thereby striking a better balance between forgetting and retaining overall performance. Variants such as SimNPO [12] removes the need for a separate reference model, and recent work such as FLAT [49] further streamlines the loss to avoid relying on D_r.

Lu *et al.* [31] proposed the Quantized Reward Konditioning method that conditions generation on discrete reward levels using special tokens; it samples outputs, bins them into reward quantiles, and fine-tunes the model to prefer high-reward outputs conditioned on their reward token, while staying close

to the original model via a KL penalty. By doing so, they are able to post-hoc unlearn undesirable properties.

More recent work like DeMem [27] utilize a negative similarity metric (between the ground truth and generation) as a reward to encourage the LLM to not memorize, ergo facilitating unlearning. While the aforementioned methods rely on end-to-end gradients with respect to specially designed loss functions, Li *et al.* [28] propose an alternative: Representation Misdirection for Unlearning (RMU) is a finetuning method that unlearns hazardous knowledge by optimizing to randomize the learned representations of D_u. As a follow up, LUNAR [40] performs unlearning by redirecting representations of forgotten data into latent regions where the model naturally signals its inability to respond. Work by Huutien *et al.* [21] also adapts RMU and steers model representation in the intermediate layer to a targeted random representation.

It is worth noting that there have also been attempts to facilitate exact unlearning to LLMs. For instance, Chowdhury *et al.* [7] also introduce a parameter-efficient fine-tuning framework that attaches lightweight, shard-specific adapter modules to the base model, enabling exact unlearning by simply removing the adapters corresponding to the data to be forgotten. But such an approach only works if the unlearning dataset is part of the fine-tuning data (and not the pre-training data).

2.3 Evaluation Metrics

Here, we taxonomize existing LLM unlearning evaluation metrics and discuss how they work. Before jumping into the details, we describe additional notation. We define $\hat{\theta}_r = \texttt{train}(\hat{\theta}_u, D')$ as the model obtained by retraining $\hat{\theta}_u$ on a dataset D'. Let $\mathcal{L}_{acc}$ denote an accuracy-based loss.

M1. Next-Token Prediction Loss: To evaluate retention of language modeling ability on forgotten content, the most straightforward approach is to measure the next-token prediction loss (which is often the training loss function of LLMs) on D_u and assert the negative of it is below a certain threshold ε,[1] *i.e.,*

$$\mathbb{E}_{(x,y)\sim D_u}[-\mathcal{L}_p(\hat{\theta}_u(x))] \leq \varepsilon_{\text{NTP}}.$$

M2. Memorization: Works in the LLM memorization literature have been used as metrics for unlearning [39]. They often measure how feasible it is to "reactivate" forgotten behaviors by adversarial prompt manipulations (i.e., like jailbreaking).

$$\mathbb{E}_{(x,y)\sim D_u}[\mathcal{L}_{acc}(x, \theta_{\text{mod}}(x))] \leq \varepsilon_{\text{mem}},$$

Following Schwarzschild *et al.* [39], we define $\theta_{\text{mod}}(x) = \hat{\theta}_u(\mathcal{A}_{\text{JB}}(x))$, where $\mathcal{A}_{\text{JB}}(x)$ is an adversarially optimized prompt that elicits x as output; that is,

$$\mathcal{A}_{\text{JB}}(x) = \arg\max_p \Pr[\hat{\theta}_u(p) = x],$$

---

[1]We add the negative sign to ensure the consistency of the meaning of the threshold – being below it indicates successful unlearning.

In other words, this metric measures the similarity between a sentence from D_u, and the generation of $\hat{\theta}_u$ when queried by a prompt that is adversarially optimized to generate the sentence. A lower metric value suggests that no adversarial prefix can generate the target sentence, indicating worse memorization and thus better unlearning, and vice versa.

M3. Relearning: For relearning-based methods [32], one retrains $\hat{\theta}_u$ on $D' \subseteq D_u$ or on out-of-distribution points to obtain a relearned model $\theta_{\mathrm{mod}} = \texttt{train}(\hat{\theta}_u, D')$. Then the next-token prediction loss is evaluated to quantify success of unlearning:

$$\mathbb{E}_{(x,y) \sim D_u}[-\mathcal{L}_p(\theta_{\mathrm{mod}}(x))] \leq \varepsilon_{\mathrm{relearn}}.$$

That is, if the model has relearned parts of D_u and still has high loss value, then it is a sign that unlearning is successful.

M4. Activation Drift: The activations of hidden layers (a.k.a., hidden states) of LLMs can also be used as a signal to infer whether the LLMs still have knowledge about D_u. If we denote the ith layer's hidden states of θ with respect to data point x as $\theta^i(x)$, then the metric would be:

$$\mathbb{E}_{(x,y) \sim D_u}[-\mathcal{L}_p(g(\hat{\theta}_u^i(x)))] \leq \varepsilon_{\mathrm{AD}}.$$

where g is a function mapping hidden states to the output space of the LLM. Motivated by an interpretation technique of LLMs named "logit lens", here g is defined to be applying the final linear output head of the LLM to the intermediate hidden states [32,36]. Intuitively, this metric can identify cases of "incorrect" unlearning, where the probability of generating the tokens related to D_u is only suppressed in later layers of the model, but not in the early layers. In other words, the knowledge about D_u is still in the parameters of the model.

M5. Membership Inference: Membership inference attacks aim to identify whether a given data point is included in the training dataset of a given trained model. It has often been used as metrics for unlearning (not restricted to LLMs). More formally, it is defined as follows:

$$\mathbb{E}_{(x,y) \sim D_u}[\mathcal{A}_{\mathrm{MI}}(\hat{\theta}_u, x)] \leq \varepsilon_{\mathrm{MIA}},$$

where $\mathcal{A}_{\mathrm{MI}}$ can be any membership inference attack for LLMs [41,53], and it outputs a membership score indicating how likely the given point x is in the training set of $\hat{\theta}_u$. Intuitively, a lower score indicates successful unlearning, since x is a training member of θ.

The metrics listed thus far are the most practical to use. We also list 2 other unlearning metrics that rely on the assumptions of either an exactly unlearned model, or on the existence of $T(D_u)$, a Q&A variant of D_u, akin to a unit test. These assumptions are hard to satisfy in practice: exact unlearning is too computationally costly for LLMs, and creating a $T(D_u)$ that ensures full coverage on D_u is challenging. In the limit, $T(D_u)$ must contain questions on all higher-order knowledge that can be inferred from D_u; this is as hard as solving unlearning in the generative setting. Thus, while we

include these metrics for completeness, due to their current impracticality, we do not consider them in our experiments.

M6. Output Distribution Similarity: When there exists an exactly unlearned model (*i.e.,* one retrained on $D - D_r$), θ_u, evaluating unlearning is trivial: one may measure how $\hat{\theta}_u$'s outputs differ from outputs of θ_u.

$$\mathbb{E}_{x \sim T(D)}[\mu_{\mathrm{KL}}(\theta_u(x), \hat{\theta}_u(x))] \leq \varepsilon_{\mathrm{OD}}.$$

We use $\mu_{\mathrm{KL}}(\cdot, \cdot)$ to denote KL divergence, since outputs of both models are probability distributions.

M7. Task Accuracy Deviation: When there exists a dataset $T(D_u)$ derived from D_u (such as prompt-response pairs, classification examples, or Q&A tasks), one is able to "test" $\hat{\theta}_u$'s knowledge about D_u on $T(D_u)$ to evaluate unlearning. Creating $T(D_u)$ can be achieved by manual efforts or neural approaches [10], and is analogous to making exams based on contents of a course, except we wish for $\hat{\theta}_u$ to perform poorly on $T(D_u)$ when unlearning is successful:

$$\mathbb{E}_{(x,y) \sim T(S(D_u))}[\mathcal{L}_{acc}(\hat{\theta}_u(x), y)] \leq \varepsilon_{\mathrm{TA}}.$$

A Common Theme: Across all metrics, a common theme is the attempt to quantify the residual influence of D_u on $\hat{\theta}_u$. Thus, we can abstract all of these unlearning metrics using $\texttt{Eval}(\hat{\theta}_u, D_u)$, *i.e.,* it is a function defined on the unlearned model and the unlearning dataset. While these methods differ, they share the goal of detecting incomplete disentanglement from the unlearning set. However, a key limitation is that D_u may be only a partial proxy for the broader property we seek to erase. If similar information is redundantly encoded in other parts of $\mathcal{D}$, removing D_u alone may be insufficient. In the rest of the paper, we propose our approach to exploit this observation.

---

3 Problem Formulation

In this section, we formally state the problem statement and the threat model. We also present a motivating experiment to provide an intuition for the limitations in existing evaluation metrics for LLM unlearning.

3.1 Problem Statement

The discussion of the metrics thus far has highlighted the main concern: *there is a narrow view of what it means to "forget" — implicitly reducing knowledge-level unlearning to the erasure of specific examples*. In training supervised-learning ML models, concerns about evaluation artifacts are generally addressed through the use of held-out validation sets that provide an unbiased measure of generalization. However, in

unlearning, no natural validation set exists: D_u is precisely the data to be removed, and datasets semantically (and distributionally) "close" to it are usually undefined/unavailable.

> **Our central objective** is to construct a surrogate validation dataset $\tilde{D}_u$ to audit the faithfulness of unlearning metrics.

This enables us to *falsify* unlearning metrics, *i.e.,* to demonstrate cases where a metric produces misleading assessments of the model's retained knowledge. We are particularly interested in scenarios where evaluation on D_u suggests unlearning is effective, whereas evaluation on our constructed $\tilde{D}_u$ suggests otherwise. We believe this is a critical failure mode as it can mean that sensitive, copyrighted, or harmful information, *intended to be removed*, still persists in deployed models.

Why Does This Matter? One might argue that forgetting D_u alone is sufficient (as that is what is specified in the unlearning objective), and that evaluation on similar but out-of-specification data (e.g., $\tilde{D}_u$) is unnecessary. However, this argument is fundamentally flawed. LLMs are designed to generalize by construction, and the influence of D_u manifests not only in "memorized" tokens (or those directly corresponding to D_u) but also in higher-level abstractions and semantic patterns. In legal and ethical contexts, such as compliance with GDPR [45], takedown of harmful content, or the removal of proprietary data, the goal of unlearning is often to erase the *knowledge* derived from D_u, not just the specific inputs. Failing to evaluate on semantically similar examples permits a model to retain and act on this learned knowledge, undermining the spirit of unlearning. By constructing and evaluating on $\tilde{D}_u$, we provide a mechanism to expose such failures and stress-test unlearning metrics. Our approach offers a more faithful audit of unlearning outcomes, ensuring that metrics reflect the true extent to which a model has relinquished knowledge associated with D_u.

3.2 Motivating Experiment

Goal: When unlearning aims to erase all knowledge and behaviors derived from D_u, a basic criterion for ideal unlearning metrics is that they yield consistent results on both the unlearning dataset D_u and a closely related dataset $\tilde{D}_u$. By consistent, we mean that the distribution of metric values on $\tilde{D}_u$ should closely match that on D_u. To test whether this criterion holds, we conduct a motivating experiment using a fantasy novel (which we refer to as Book X) as D_u and its fan fiction as $\tilde{D}_u$.
Approach & Setup: We collect a dataset of 12 fantasy novels and their corresponding fan fiction, where Book X is one of the 12 novels, and fine-tune a Llama-3-8B [15] model on them to simulate the system requiring unlearning. Book X is of particular interest because it was published after the model's pre-training cutoff date. This ensures that the base model
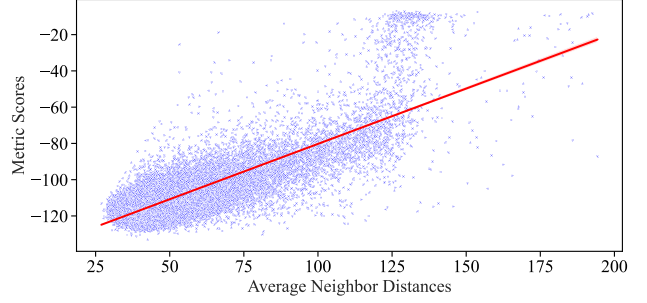

Figure 1: **Min-k% metric scores with respect to the average embedding distance from data points in $\tilde{D}_u$ to the 100 nearest neighbors in D_u.** One can see a clear correlation as indicated by the red regression line in the figure. This suggests despite the metric is expected to perform consistently across D_u and $\tilde{D}_u$, it can be impacted by where the data points located in the embedding space.

has not been exposed to it during pre-training, allowing us to disentangle unlearning effects from confounders introduced by pre-existing. We then unlearn D_u using NPO [54], treating the remaining novels (excluding fan fiction) as the retain dataset to preserve model utility. Once the unlearned model $\hat{\theta}_u$ is obtained, we compute an unlearning metric from category M5, Min-k% [41], on $\tilde{D}_u$ (more details in § 5).

To evaluate whether the metric behaves consistently across D_u and $\tilde{D}_u$, we take each sentence from $\tilde{D}_u$ and measure its embedding similarity to sentences from D_u. Our hypothesis is that if the metric is indeed consistent, the metric value for a sentence in $\tilde{D}_u$ should be independent of its embedding similarity to sentences in D_u. Specifically, we use the last hidden states of the model (outputs of the final transformer block averaged across the sequence length) as embeddings, where the embedding of a point x is denoted by $e(x)$. We use $\mu(\cdot, \cdot)$ to denote a certain similarity or distance metric, such as KL divergence, Jensen–Shannon divergence, ℓ_2 distance, or cosine similarity. In our analysis, we adopt ℓ_2 distance:

$$\mu(x \sim D_u, y \sim \tilde{D}_u) = \|e(x) - e(y)\|.$$

For each sentence from $\tilde{D}_u$, we identify its 100 nearest neighbors in D_u using this distance and plot the Min-k% scores of the corresponding sentence against these distances (Figure 1).
Results: It can be observed that the two axes are highly correlated: sentences from $\tilde{D}_u$ that lie further from the embedding distribution of D_u tend to receive lower unlearning scores, *i.e.,* they are more likely to be judged as "not unlearned." At first glance this may seem reasonable: embeddings are often assumed to capture semantic meaning, so one might expect that sentences semantically distant from D_u be considered irrelevant for unlearning. However, semantic distance alone does not imply the absence of information from D_u. In our case, $\tilde{D}_u$ consists of fan fictions of D_u; although semantically

different, they are directly related to and built upon D_u. This contradicts our hypothesis and shows that the metric under consideration fails to behave consistently across D_u and $\tilde{D}_u$.

This raises a broader concern: *does a dataset analogous to $\tilde{D}_u$ always exist, independent of the model, dataset, and unlearning algorithm, and for all proposed unlearning metrics? If so, then no metric could ever definitively certify successful unlearning, since there would always be points highly related to D_u yet assigned values suggesting the opposite.*

3.3 Requirements & Threat Model

To address the question raised by the motivating experiment on falsifying unlearning metrics, we now formally state the requirements and the threat model. Any entity aiming to falsify the effectiveness of a metric would need to demonstrate the existence of a dataset $\tilde{D}_u$ such that:

1. $\tilde{D}_u$ is provably closely related to, or can be learned from, D_u, but

2. The metric values computed on $\tilde{D}_u$, $\texttt{Eval}(\hat{\theta}_u, \tilde{D}_u)$, are inconsistent with $\texttt{Eval}(\hat{\theta}_u, D_u)$.

To achieve these two criteria, we believe the following properties are necessary for $\tilde{D}_u$.

- **P1. Natural Semantic Relevance**: *The dataset should be meaningfully related to D_u and expressed in natural language.* $\tilde{D}_u$ must be both semantically related to D_u and linguistically fluent. This rules out adversarially crafted datasets that elicit specific responses but consist of meaningless or ungrammatical text, as such datasets are neither naturally occurring nor informative for evaluating unlearning. We quantify this property in two ways: (i) by comparing the likelihood of a sentence under the model before unlearning versus after finetuning on D_u, where a significant increase indicates related knowledge, and (ii) by measuring fluency using standard language-model metrics such as perplexity.
- **P2. Embedding Separation**: *The dataset should stress-test metrics by including semantically close but embedding-wise distant cases.* As motivated in § 3.2, samples from $\tilde{D}_u$ that are semantically close to D_u but far apart in the embedding space are particularly informative. Unlearning metrics often behave inconsistently in such cases (as observed in the fan fiction dataset), making them critical boundary tests. By including these samples, we can examine whether metrics only work in the immediate neighborhood of D_u or remain robust to embedding distance when semantic relevance is preserved. This property can be quantified by measuring the embedding distance from a sample to its k nearest neighbors in D_u.

What Does This Mean? Intuitively, the existence of $\tilde{D}_u$ (that satisfies the properties listed above) implies two things: (1) $\texttt{Eval}$ is not an accurate or comprehensive evaluation mechanism for unlearning. In particular, assume $\texttt{Eval}$ belongs to

one of the metric categories defined in § 2.3, and that its values lie in $\mathbb{R}^+$. Then, demonstrating $\texttt{Eval}(\hat{\theta}_u, \tilde{D}_u) > \texttt{Eval}(\hat{\theta}_u, D_u)$ shows that the metric can produce false positives *i.e.,* claiming unlearning is successful when it is not. (2) The metric $\texttt{Eval}$ can be exploited to spoof successful unlearning by selectively reporting results on a subset of $\tilde{D}_u$ that yields better scores than D_u. This latter case is particularly relevant when the full D_u cannot be revealed, for example in scenarios involving unlearning bio-weapon–related knowledge.

Threat Model: We consider an *auditor i.e.,* an entity tasked with evaluating unlearning metrics on behalf of an external authority (such as a regulator), so that the metrics may be used by the external authority to faithfully assess knowledge-based unlearning. Since the focus is on unlearning metrics rather than the process of unlearning in this threat model, the auditor may use arbitrary models and unlearning datasets. Thus, the auditor has white-box access to the model before unlearning θ, the model after unlearning $\hat{\theta}_u$, and the unlearning dataset D_u. The auditor may lack sufficient computational resources to perform exact unlearning (*i.e.,* retraining from scratch), but can carry out lightweight computations such as fine-tuning. White-box access also enables the auditor to inspect hidden states of the model. For example, the auditor can compute sentence embeddings of data points by averaging token embeddings from hidden layers, denoted $e_\theta(\cdot)$, where θ is the model under consideration.

4 Proximal Surrogate Generation (PSG)

The objective of our algorithm is to generate natural-language sentences that contain information derived from D_u, yet exhibit large embedding distances from the actual sentences in D_u (recall the criteria described in § 3.3). The underlying hypothesis is that current unlearning metrics are unreliable because they are too dependent on embedding-space distance from D_u. Yet, as observed in § 3.2, embedding-space distance is not a reliable proxy for measuring whether two sets of sentences share information. By generating sentences derived from D_u, yet are purposely far from D_u in the embedding-space, we can test this hypothesis: if a metric certifies a model to have successfully unlearned sentences in D_u, but not the derived sentences that contain knowledge inferred from D_u, it is likely not reliable because it is overly sensitive to embedding-space distance.

To achieve the objective, we propose Proximal Surrogate Generation (PSG), which we formalize in Algorithm 1.

4.1 Algorithm Overview

Given an LLM θ and an unlearning dataset D_u, we fine-tune θ on D_u to obtain θ_f (**line 2** of Algorithm 1).[2] We use θ_f to generate $\tilde{D}_u$, a set of natural-language sentences that are derived

---

[2]A concern the reader may have here is that the LLM could have already been exposed to D_u during pre-training, and thus have a high performance

Algorithm 1 Proximal Surrogate Generation (PSG)

Require: Original model θ, unlearning dataset D_u, iterations N, embedding function $e(\cdot)$, distance metric $\mu(\cdot,\cdot)$
Ensure: $\tilde{D}_u$

1:   $\tilde{D}_u = \{\}$
2:   $\theta_f \leftarrow SFT(\theta, D_u)$ $\triangleright$ Finetune θ on D_u
3:   $E_f = \{e(x) | \forall x \in D_u\}$ $\triangleright$ Compute embedding for every sentence in D_u
4:   **for** $sent \in D_u$ **do**
5:       $x = generate_\theta(sent)$ $\triangleright$ Generate the next sentence
6:       **for** $t = 1$ to N **do**
7:           $x^* \leftarrow \arg\min_x -w_{\text{dist}} d(e(x), minibatch(E_f)))$ $\triangleright$ **Maximize Embedding Distance** via GCG
8:           $x_{prompt} \leftarrow$ "Rephrase: " $+ x^* +$ " Rephrased: " $\triangleright$ Prompt for Rephrasing
9:           $S \leftarrow w_{\text{prefer}} \Pr_{\theta_f}(. | x_{prompt}) + w_{\text{likelihood}} \Pr_\theta(. | x_{prompt})$ $\triangleright$ Weighted Summed Objective
10:         $x \leftarrow \text{sample}(S, \theta)$ $\triangleright$ **Sampling-based rephrasing**
11:       $x_{\text{final}} \leftarrow x$
12:       **if** $\Pr_{\theta_f}(x_{\text{final}}) - \Pr_\theta(x_{\text{final}}) > \tau_{\text{prefer}}$ & $d(e(x_{\text{final}}), KNN(e(x_{\text{final}}), E_f))) > \tau_{\text{dist}}$ & $\Pr_\theta(x_{\text{final}}) > \tau_{\text{likelihood}}$ **then** $\triangleright$ **Filtering**
13:           $\tilde{D}_u \leftarrow \tilde{D}_u \cup \{x\}$ $\triangleright$ Append sentences satisfying G1, G2, and G3
14:   **return** $\tilde{D}_u$

---

from D_u, yet are for from sentences in D_u in the embedding space. To facilitate this, we define an *objective function* that combines likelihood increase with the standard likelihood objective for sampling (**line 9**), and a complementary *loss function* that penalizes proximity to D_u in the embedding space (**line 7**). To generate sentences for $\tilde{D}_u$, we iteratively rephrase every sentence in D_u to maximize the objective (ensuring **P1**) while minimizing the loss (ensuring **P2**). The former is achieved through sampling (**line 10**), while the latter uses a GCG-like optimization step (**line 7**). Unlike standard GCG, which forces the generation of predefined outputs [55], our variant allows the model to generate any text as long as it increases embedding distance from D_u. This process can be seen as creating adversarial examples to maximize embedding separation, with fluency preserved through filtering (**line 12**). This filtering stage is applied after N iterations, based on three requirements: (1) the likelihood increase from θ to θ_f, which captures semantic relevance (and excludes gibberish by ensuring the sentence has sufficient likelihood under θ); (2) the embedding distance to the k nearest neighbors from D_u, which ensures embedding separation; and (3) the likelihood of the sentence with respect to θ for fluency, as aforementioned. To support verifying the second criterion efficiently, we pre-compute embeddings of all sentences in D_u using θ (**line 3**), avoiding repeated encoding during distance calculations. Sentences that exceed user-defined thresholds on all three axes (with guidelines in § 4.2 and § 4.3) are retained as natural candidates derived from D_u that may falsify unlearning metrics, composing the surrogate dataset $\tilde{D}_u$. That said, differing

from *e.g.,* Q&A datasets used to evaluate unlearning, $\tilde{D}_u$ is not meant to cover all derivable knowledge from D_u. Instead, it is designed for a falsification-style stress test, where the sentences contained by it are "boundary cases" that likely have different metric values from D_u. Therefore, $\tilde{D}_u$ does not need to be a large dataset but can be as small as *e.g.,* 100 sentences, meaning PSG is not computationally expensive.

Finally, it is worth noting that PSG requires only the dataset D_u and the pre-unlearning model θ as input; it does not depend on the unlearned model $\hat{\theta}_u$.

4.2 Achieving Natural Semantic Relevance

The **P1** criterion is enforced through both the sampling and filtering stages of Algorithm 1. To approximate how the model generalizes from D_u, we fine-tune the original pre-unlearning model θ solely on D_u, obtaining θ_f. By definition, if training on D_u reduces the cross-entropy loss for next-token prediction on a data point outside D_u, then the model has generalized from D_u to that point. In unlearning scenarios motivated by, *e.g.,* copyright or safety concerns, such points should also be unlearned. Accordingly, we would expect unlearning metric values on these points to align with those obtained directly on D_u. This captures our key intuition: *one needs to leverage observations from learning to better understand unlearning.*

However, the reverse does not always hold: not all points influenced by D_u will show a measurable loss decrease during fine-tuning. This is because D_u was already part of θ's training, and some points may have plateaued at low loss. To bias sampling toward sentences that plausibly reflect generalization, we adapt the standard likelihood objective for token generation. In particular, given a prompt x_{prompt}, the model's token distribution is adjusted as

$$w_{\text{prefer}} \Pr_{\theta_f}(\cdot \mid x_{\text{prompt}}) + w_{\text{likelihood}} \Pr_\theta(\cdot \mid x_{\text{prompt}}),$$

---

on knowledge generalized from D_u even without fine-tuning. While it is possible to pre-train from scratch to avoid this, the computational costs make this option impractical. The key observation is that PSG only depends on θ improving its performance on D_u after fine-tuning relative to before, which we found to be the case in all experiments.

with $w_{\text{prefer}} > w_{\text{likelihood}}$ and the weights normalized to sum to 1. This ensures preference for tokens whose likelihood increases from θ to θ_f, while still preserving the (pretrained) base model's fluency. Sampling under this mixed objective therefore produces candidates that are both semantically tied to D_u and linguistically natural.

At the end of the rephrasing loop, we obtain a candidate x_{final} that has also undergone a few iterations of embedding-distance maximization (see next subsection). Because this step may cause the sentence to drift away from semantic relevance, we apply a final filtering criterion. Specifically, x_{final} is retained in $\tilde{D}_u$ only if

$$\Pr_{\theta_f}(x_{\text{final}}) - \Pr_{\theta}(x_{\text{final}}) > \tau_{\text{prefer}},$$

where τ_{prefer} is a threshold. As a heuristic, it may be set based on the likelihood increase from θ to θ_f across D_u. For example, in the rest of this paper, we set it to the first quartile of this likelihood increase. This ensures that only sentences relevant to D_u can be included in $\tilde{D}_u$, since irrelevant sentences would have an unchanged or even decreased (due to catastrophic forgetting [14]) likelihood when the model is only finetuned on D_u. In addition, we also require that the likelihood under θ itself exceeds $\tau_{\text{likelihood}}$. To discard low-quality or ungrammatical sentences, $\tau_{\text{likelihood}}$ may be set to the first quartile of likelihoods on D_u so sentences in $\tilde{D}_u$ are comparably fluent to the ones from D_u. Together, these filtering conditions ensure that sentences in $\tilde{D}_u$ remain both semantically related to D_u and fluent.

4.3 Achieving Embedding Separation

To achieve **P2**, *i.e.,* encouraging large embedding distances between points created by PSG and the unlearning dataset D_u, we adapt a GCG-like optimization [55]. GCG is commonly used as a jailbreaking method: the model is queried with an adversarial prompt consisting of "free tokens" and a malicious or toxic request that would otherwise be refused. The free tokens are then optimized to maximize the likelihood of the model producing a specific continuation, *e.g.,* `"Here is the response..."`, thereby coercing it into answering the malicious prompt. In our setting, the goal is to maximize embedding distance rather than induce a specific output (see Appendix B for details). We therefore modify the GCG loss to

$$-w_{\text{dist}}\, d\big(e(x), \text{minibatch}(E_f)\big),$$

where $e(\cdot)$ denotes the embedding function, E_f is the set of embeddings for D_u, and $d(\cdot, \cdot)$ is the averaged chosen distance metric (Euclidean or cosine). Performing GCG to minimize this loss increases the embedding distance. Concretely, the embedding function $e(\cdot)$ is the mean of the last-layer hidden states of θ (the model before unlearning), and E_f can be precomputed before entering the for-loop in Algorithm 1 to reduce computation. Distances are computed as the average

distance from $e(x)$ to a mini-batch sampled from E_f. The optimization produces a pair $(x^*, \text{free tokens})$, analogous to adversarial prompts in standard GCG. When used in the rephrasing stage, querying the model with x^* yields outputs that are distant from E_f in the embedding space.

Finally, in the filtering stage, we discard points with insufficient embedding-distance increase, using a threshold τ_{dist}. Here, instead of computing the embedding distance from x_{final} to a random minibatch sampled from D_u, we find the k nearest neighbors (e.g., $k = 100$) of x_{final} from D_u, to reduce stochasticity. Then, in practice, τ_{dist} may be set to the embedding distance from a point in D_u to its k nearest neighbors, averaged across all points in D_u, multiplied by a coefficient > 1. This ensures the sentences included in $\tilde{D}_u$ to have higher embedding distances to D_u than an average point in D_u, preventing memorized patterns from D_u to be included in $\tilde{D}_u$.

Why GCG? One may ask why a GCG-like method is necessary instead of the sampling-based approach used in the previous subsection. The key reason is that sampling operates on probability distributions over the vocabulary: a natural output when querying an LLM. In contrast, selecting the token that maximizes embedding-distance increase would require computing embeddings for the entire vocabulary. Such embeddings are not a standard output of LLM queries, and given the typically large vocabulary size, this would incur an intractable computational cost.

5 Experimental Setup

Models, Datasets & Algorithms. We conduct experiments on a combination of 3 pretrained LLMs: Meta-Llama-3-8B [15], Qwen2.5-7B [51], and Zephyr-7B-Beta [44], together with 3 unlearning datasets from distinct domains: (i) a fantasy novel denoted as Book X[3], (ii) a toxic subset of comments from the Civil Comments platform [3], and (iii) a dataset containing biological weapon–related knowledge, WMDP-bio [28]. We apply 2 representative unlearning techniques: NPO [54] and RMU [28]. Combining 3 LLMs, 3 unlearning datasets, and 2 techniques yields 18 unlearned models in total. Unless otherwise specified, the same hyperparameters of PSG are used across all settings.

For each triplet (LLM, unlearning dataset D_u, unlearning technique), we first fine-tune the LLM on D_u combined with additional datasets from similar distributions.[4] Portions of these additional datasets serve as retain sets during unlearning, enabling us to simulate an LLM trained on D_u without training from scratch. We then perform unlearning using the corresponding retain set to preserve model utility. The full configuration of unlearning and retain datasets for each ex-

---

[3]We use this anonymized novel instead of the more commonly considered *Harry Potter* because the latter may not represent typical copyrighted content, given its extensive presence across English datasets.

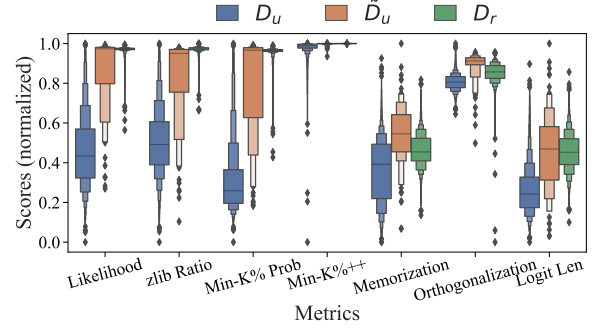[4]An exception is WMDP-bio, which prior work has shown to already be included in pretraining corpora [28].

periment is summarized in Table 2, with additional details (including compute resources) provided in Appendix A.

Unlearning Metrics Used. After obtaining the unlearned models, we evaluate unlearning success using the metrics described in § 2.3. In particular, we considered 7 metrics and their implementation details are as follows:
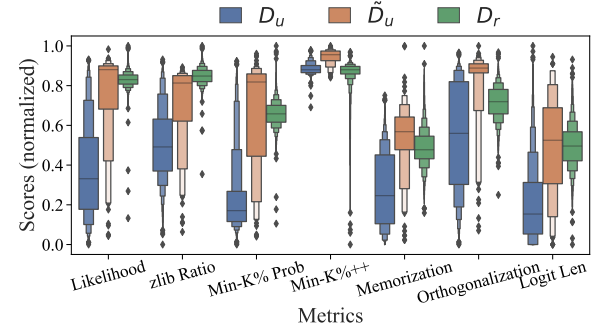
- *Likelihood* (M1) is directly computed from the model's loss value, *i.e.,* the negative log-likelihood of a given sentence.
- *Memorization* (M2) is a variant of Adversarial Compression Ratio (ACR) [39], a metric for evaluating memorization in LLMs. ACR measures the number of adversarial tokens required by GCG to force the model to reproduce an exact training point. However, varying the number of adversarial tokens makes this method computationally expensive for large datasets. We therefore modify it to instead measure the number of identical tokens between the model's generation and a training sentence, given a fixed number of adversarial tokens.
- *Orthogonalization* [32] (M3) is a form of relearning attack. It uses a small fraction of D_u to estimate the parameter change caused by unlearning, and then applies this change back to the model. The likelihood of sentences under this modified model serves as a metric for how easily the unlearned knowledge can be relearned.
- *Logit Lens* [36] (M4) examines the hidden states of LLMs. Most current LLMs consist of several transformer blocks followed by a linear classification head to predict the next token. The hypothesis behind this metric is that unlearned content may only be suppressed in later layers. Accordingly, the method connects the linear head to an earlier transformer block to predict the next token. The likelihood of sentences under this setup defines the metric value.
- *zlib Ratio* (M5) is the ratio between likelihood and the zlib entropy [13] of a sentence. It was proposed as a membership inference technique for LLMs [6]. The idea is that a sentence is more likely to be a member if it has low entropy, which can result from, *e.g.,* repeated patterns of memorized training text.
- *Min-K% Prob* (M5) is another widely used membership inference technique for LLMs [41]. Its intuition is that for a sentence seen during training, even its lowest-probability tokens should exceed a certain threshold.
- *Min-K%++* (M5) is a follow-up [53] to Min-K% Prob. The main difference is that it normalizes per-token probability by the expected probability of all tokens in the vocabulary, since tokens observed during training typically have locally (rather than globally) maximal probability.

6 Evaluation

In this section, we design experiments to simulate scenarios where trained LLMs are required to unlearn parts of their training datasets. Our evaluation is guided by 3 questions:

(a) Negative Preference Optimization (NPO)

(b) Representation Misdirection for Unlearning (RMU)

Figure 2: **Boxplots of metric scores of the unlearning dataset D_u, surrogate unlearning dataset $\tilde{D}_u$, and retain dataset D_r, when *Book X* is unlearned from a Llama-3 model using (a) NPO and (b) RMU.** The metric scores are normalized to a [0,1] range, and a larger value indicates unsuccessful unlearning (according to the metrics). One can observe that the box corresponding to $\tilde{D}_u$ is almost always higher than the box corresponding to D_u. This indicates that for this setting, PSG is able to create sentences that falsify the unlearning metrics by demonstrating they are unlearned less successfully than D_u.

- How do existing unlearning metrics behave when applied to datasets that are surrogates of the unlearning dataset?
- Are the inconsistencies we observe specific to a single model–dataset pair, or do they generalize across models, datasets, and unlearning methods?
- How sensitive is PSG to its design choices, such as embedding separation thresholds, distance metrics, and generation strategies?

  Our empirical findings can be summarized as follows:

- Surrogate datasets created by PSG consistently yield metric values that differ substantially from those of the original unlearning datasets, despite being semantically tied to them (as shown in the case study in § 6.1.)
- These inconsistencies generalize broadly: across 3 models, 3 datasets, and 2 unlearning methods, no existing met-

Dataset	Model	Method	Unlearning Metrics						
			Likelihood	zlib Ratio	Min-K% Prob	Min-K%++	Memorization	Orthogonalization	Logit Len
Book X	Llama-3	NPO	**2.13 +/- 0.16**	**1.61 +/- 0.12**	**2.16 +/- 0.14**	**0.90 +/- 0.19**	**1.16 +/- 0.09**	**1.37 +/- 0.07**	**1.20 +/- 0.09**
		RMU	**1.56 +/- 0.12**	**0.93 +/- 0.10**	**1.74 +/- 0.14**	**1.68 +/- 0.10**	**1.35 +/- 0.11**	**1.13 +/- 0.11**	**1.18 +/- 0.11**
	Qwen2.5	NPO	**2.01 +/- 0.12**	**0.54 +/- 0.09**	**1.96 +/- 0.13**	**1.53 +/- 0.11**	**1.51 +/- 0.09**	-0.49 +/- 0.06	**0.82 +/- 0.08**
		RMU	**1.47 +/- 0.13**	**0.77 +/- 0.12**	**1.57 +/- 0.12**	**1.43 +/- 0.15**	**1.62 +/- 0.10**	**1.54 +/- 0.13**	**0.85 +/- 0.10**
	Zephyr	NPO	**1.47 +/- 0.09**	0.20 +/- 0.05	**1.42 +/- 0.08**	**1.33 +/- 0.09**	**1.18 +/- 0.09**	-0.35 +/- 0.04	**1.03 +/- 0.07**
		RMU	**1.36 +/- 0.10**	-0.11 +/- 0.04	**1.30 +/- 0.09**	**1.14 +/- 0.07**	**1.18 +/- 0.08**	**1.08 +/- 0.11**	**0.98 +/- 0.08**
Toxic Civil Comments	Llama-3	NPO	**0.65 +/- 0.10**	0.23 +/- 0.09	**0.73 +/- 0.10**	**0.54 +/- 0.08**	**0.89 +/- 0.07**	-0.23 +/- 0.06	0.46 +/- 0.09
		RMU	**0.67 +/- 0.11**	-0.10 +/- 0.08	**0.66 +/- 0.09**	**0.62 +/- 0.08**	**0.79 +/- 0.08**	**0.87 +/- 0.10**	0.44 +/- 0.09
	Qwen2.5	NPO	0.38 +/- 0.08	-0.15 +/- 0.02	**0.56 +/- 0.09**	0.40 +/- 0.09	**0.62 +/- 0.07**	0.06 +/- 0.04	0.16 +/- 0.10
		RMU	**0.70 +/- 0.08**	-0.23 +/- 0.05	**0.78 +/- 0.09**	**1.04 +/- 0.09**	**0.97 +/- 0.07**	0.40 +/- 0.05	-0.02 +/- 0.09
	Zephyr	NPO	0.15 +/- 0.09	**-0.51 +/- 0.05**	0.26 +/- 0.10	0.26 +/- 0.10	0.29 +/- 0.07	0.40 +/- 0.08	0.04 +/- 0.08
		RMU	0.05 +/- 0.09	**-0.63 +/- 0.03**	0.19 +/- 0.10	-0.15 +/- 0.08	0.27 +/- 0.07	-0.07 +/- 0.09	-0.04 +/- 0.08
WMDP-bio	Llama-3	NPO	0.49 +/- 0.05	**-1.06 +/- 0.02**	0.25 +/- 0.03	0.23 +/- 0.04	**0.88 +/- 0.05**	**0.65 +/- 0.09**	0.36 +/- 0.02
		RMU	**0.88 +/- 0.03**	**-1.10 +/- 0.02**	0.46 +/- 0.06	0.07 +/- 0.08	**0.89 +/- 0.05**	**-4.10 +/- 0.03**	0.39 +/- 0.00
	Qwen2.5	NPO	0.34 +/- 0.04	**-1.30 +/- 0.01**	0.24 +/- 0.04	-0.15 +/- 0.00	**1.08 +/- 0.04**	**3.44 +/- 0.04**	**0.55 +/- 0.04**
		RMU	**0.66 +/- 0.03**	**-1.42 +/- 0.02**	0.24 +/- 0.05	-0.46 +/- 0.06	**1.27 +/- 0.05**	**2.32 +/- 0.18**	0.48 +/- 0.01
	Zephyr	NPO	0.28 +/- 0.07	**-1.56 +/- 0.04**	0.33 +/- 0.07	0.37 +/- 0.07	**0.52 +/- 0.09**	0.37 +/- 0.10	0.30 +/- 0.03
		RMU	**0.98 +/- 0.09**	**-0.92 +/- 0.04**	**0.90 +/- 0.08**	**1.02 +/- 0.08**	**1.02 +/- 0.11**	0.45 +/- 0.11	0.45 +/- 0.05

Table 1: **Standardized mean difference between metric values of $\tilde{D}_u$ and D_u across various settings.** For each combination of the 3 datasets, 3 models, and 2 unlearning methods, we compute the 7 metrics on the unlearned model using both $\tilde{D}_u$ and D_u. Then for each $\tilde{D}_u$ (100 points), we randomly sample 100 points from D_u and measure the mean difference between their metric values, standardized by the pooled standard deviation. The random sampling is repeated 100 times so that we report average mean differences with their standard deviations in this table. We presenting the mean difference with absolute value greater than 0.5 in bold—this is 61.1% of the table. In addition, no metric performs consistently on D_u and $\tilde{D}_u$ under all settings.

ric can always remain consistent between D_u and $\tilde{D}_u$ (as shown in § 6.1.)

- Ablation studies show that PSGdoes not require heavy hyperparameter tuning and behaves predictably with respect to its hyperparameters: larger separation thresholds increase effectiveness but reduce efficiency while choices of embedding distance metrics and generation strategies have insignficiant impacts (as shown in § 6.2.)

6.1 Achieving Falsification

Case Study: Fantasy Novel Unlearning. We begin with the scenario where the fantasy novel Book X is unlearned from a Llama-3 model using NPO. We construct the surrogate unlearning dataset $\tilde{D}_u$ and compute the values of 7 unlearning metrics, alongside those for the unlearning dataset D_u and the retain dataset D_r. To visualize the results, we plot the distributions of the metric values as boxplots in Figure 2. Since different metrics operate on different scales, we normalize each metric by its minimum and maximum values across the 3 datasets, mapping them into the range [0, 1], where larger values indicate less successful unlearning. It is noteworthy that the variability of metric values is large for some metrics, even for D_u. However, this is not problematic as long as the values computed on D_u are significantly lower than D_r, meaning the differences between the unlearned data and retaining data can be captured by the corresponding metric. *Here, this case study serves as a concrete starting point to illustrate how $\tilde{D}_u$ behaves relative to D_u and D_r.*

Divergence Between $\tilde{D}_u$ and D_u. As expected, the boxes corresponding to D_u lie below those of D_r, indicating that unlearning impacts the model's behaviors on D_u, which is successfully captured by the metrics. However, the boxes corresponding to $\tilde{D}_u$ created by PSG are markedly different from those of D_u. This is noteworthy because we set $\tau_{\text{prefer}} = \text{Quantile}_{0.25}\left(\text{Pr}_{\theta_f}(x) - \text{Pr}_{\theta}(x) \mid x \in D_u\right)$, which ensures that every sentence in $\tilde{D}_u$ exhibits a likelihood increase greater than at least 25% of the points in D_u. This guarantees that $\tilde{D}_u$ contains information and knowledge learned from D_u. Yet, the metric values suggest that $\tilde{D}_u$ is not unlearned as effectively. In other words, *these results show that $\tilde{D}_u$, despite being semantically tied to D_u, produces very different outcomes under existing metrics.*

Results Across Models and Datasets. To validate the consistency of these findings, we run PSG for all other dataset–model pairs to obtain $\tilde{D}_u$, using the same hyperparameters, and evaluate the same 7 unlearning metrics on both D_u and $\tilde{D}_u$ with respect to the corresponding unlearned models.[5] Since an ideal unlearning metric should behave similarly on D_u and $\tilde{D}_u$, we compute standardized mean differences between their metric values, reported in Table 1 (boxplots similar to Figure 2 for these settings can be found in Figures 8

---

[5] A related concern is whether the gap between the metric scores for D_u and $\tilde{D}_u$ may be a result of variances in the unlearning process, *i.e.,* the gap only exists for some unlearning runs, but not others. To test this, we run NPO on Llama-3 with the Book X unlearning set 4 times with identical hyperparameters but different random seeds (Figure 7); results show the gap between D_u and $\tilde{D}_u$ is consistent across all runs and metrics.
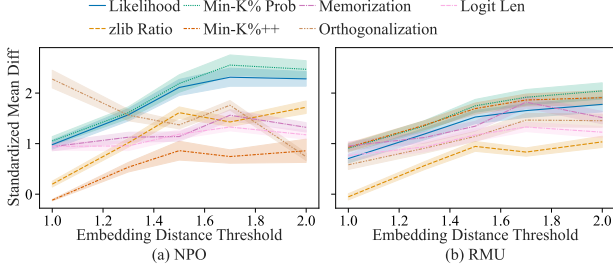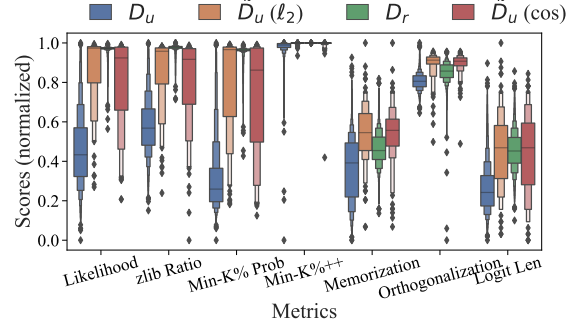
Figure 3: **Standardized mean difference between metric values of D_u and $\tilde{D}_u$, with respect to τ_{dist}.** Here τ_{dist} is relative to the average embedding distance among points in D_u. It can be observed that for both NPO and RMU, there is a positive correlation between the standardized mean difference and τ_{dist} for most of the metrics. This validates our hypothesis that increasing the embedding distances between D_u and $\tilde{D}_u$ can cause the metrics to perform more differently on them.

to 15 in Appendix C). Following standard practice, we bold the cells with absolute values exceeding 0.5, indicating cases of inconsistency. We observe that this occurs in the majority of cases, and no metric maintains a mean difference below 0.5 across all dataset–model pairs. These results validate our hypothesis from § 3: most existing unlearning metrics depend heavily on the embedding-space closeness of the evaluation points to D_u. In other words, the metrics overfit to D_u and fail to reliably judge unlearning success on sentences that are semantically derived from D_u. *Taken together, this highlights a systemic limitation: current unlearning metrics are not robust to semantically related but embedding-distant data.*

6.2 Ablation Study

PSG involves several tunable hyperparameters, as well as design choices such as the distance metric used for computing embedding separation. In this section, we conduct ablation studies to examine the effects of these hyperparameters and justify our design decisions.

Embedding Separation Threshold. As discussed in § 4, one essential component of PSG is to maximize the separation between sentences in $\tilde{D}_u$ and those in D_u. This is achieved by the GCG-like optimization and the filtering stage (lines 7 and 12 in Algorithm 1, respectively). As aforementioned, the filtering threshold τ_{dist} may be set to the average embedding distance from a point in D_u to its k nearest neighbors, multiplied by a coefficient. Doing so avoids the need to search for hyperparameters for different model-dataset pairs. So far, we used $k = 100$ and the same coefficient 1.5 for all the settings. Here, to study the impact of embedding separation on the performance of PSG, we run PSG with varying coefficient values ranging from 1 to 2 and plot the standardized mean difference

(a) Negative Preference Optimization (NPO)

(b) Representation Misdirection for Unlearning (RMU)

Figure 4: **Boxplots of metric scores of 2 surrogate unlearning datasets, $\tilde{D}_u(\ell_2)$ and $\tilde{D}_u(cos)$, where the embedding distance is measured using ℓ_2 distance and cos distance respectively.** The metric scores of the unlearning dataset D_u and retain dataset D_r are also presented as reference. One can observe that for most cases, ℓ_2 distance results in slightly more different (mean) scores from D_u than cos distance, and smaller variance in the distribution of the metric scores. Although the differences are not significant, we choose to use ℓ_2 distance as the result of this ablation study.
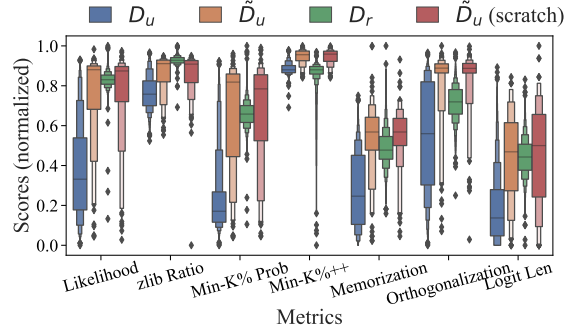
between unlearning metric values of D_u and $\tilde{D}_u$. Results for Llama-3 are shown in Figure 3, while those for Qwen2.5 and Zephyr appear in Figures 16 and 17 in Appendix C.

As seen in the figures, there is a clear positive correlation between τ_{dist} and the standardized mean difference (which represents the effectiveness of PSG). When the coefficient is 1, *i.e.,* τ_{dist} equals the average embedding distance among points in D_u, standardized mean differences are generally below 1, with some near 0, indicating little separation between D_u and $\tilde{D}_u$. As τ_{dist} increases, standardized mean differences rise accordingly. That said, since τ_{dist} is used for filtering, higher thresholds reduce the number of unfiltered sentences and thus slow down PSG when constructing $\tilde{D}_u$. *In short, larger separation thresholds make PSG more effective but at the cost of efficiency.*

Distance Metric. Beyond the magnitude of separation, the

(a) Negative Preference Optimization (NPO)

(b) Representation Misdirection for Unlearning (RMU)

Figure 5: **Boxplots of metric scores of 2 surrogate unlearning datasets, $\tilde{D}_u$ and $\tilde{D}_u$(scratch).** The sentences in the former are generated by continuing writing from sentences in D_u, whereas the latter contains sentences written from scratch (*i.e.,* generating from an empty string). The metric scores of the unlearning dataset D_u and retain dataset D_r are also presented as reference. It can be seen that for most of the metrics, the difference between the boxes corresponding to the 2 surrogate unlearning datasets are negligible. Thus, we conclude that generating whether from a sentence in D_u or an empty string does not significantly impacts the performance of PSG.

choice of distance metric is another hyperparameter of PSG. We consider 2 common options: Euclidean (ℓ_2) distance and cosine distance. To test whether this choice significantly affects PSG, we create $\tilde{D}_u$ using each metric and plot their corresponding metric values in Figure 4, following the same format as Figure 2. Two observations emerge: (1) when cosine distance is used, the resulting metric distributions tend to have larger variance than with ℓ_2 distance; (2) for some metrics, ℓ_2 distance yields larger standardized mean differences relative to D_u. However, this is not consistent across all metrics, and the differences are generally subtle. For consistency, we use ℓ_2 distance in other experiments, though PSG appears relatively robust to the choice of metric. *Overall, PSG is not highly sensitive to whether cosine or Euclidean distance is used.*

Generation Strategy. As shown in lines 4–5 of Algorithm 1, PSG begins with sentences from D_u, generates continuations, and then enters the loop of maximizing separation and rephrasing. A natural question is whether starting from D_u is necessary. To test this, we consider an alternative strategy where PSG generates sentences from scratch by prompting the model with an empty string. We denote the resulting dataset as $\tilde{D}_u$(scratch), and plot its metric values in Figure 5, alongside those of D_u, D_r, and the $\tilde{D}_u$ created by continuing from D_u. The figure shows that $\tilde{D}_u$(scratch) and $\tilde{D}_u$ yield very similar metric distributions across most metrics and both unlearning methods. This suggests that the choice between continuing from D_u or generating from scratch has little effect, likely because the iterative rephrasing loop heavily modifies sentences after initialization. Nonetheless, we prefer the former strategy, since generation from scratch can lead to collisions; for some models, starting from an empty string can result in repeated outputs with high probability. *Thus, while either generation works in principle, continuing from D_u is more practical.*
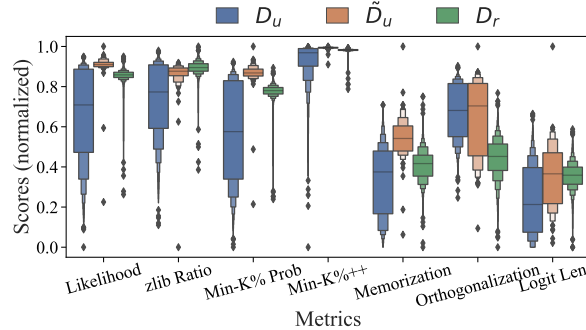
Model Size. All models studied so far have around 7–8 billion parameters (see § 5). To assess the impact of model size, we test PSG on a smaller model: Phi-3-mini-4k-instruct [1], with only 3.8 billion parameters. Other than the architecture, all settings remain the same as in Figure 2: the model is fine-tuned with 12 fantasy novels and their fan fictions, and Book X is unlearned using both NPO and RMU. The metric values of D_u, D_r, and $\tilde{D}_u$ are shown in Figure 6. Two main observations arise: (1) unlearning appears less successful for the smaller model, as D_u often overlaps with D_r in metric distributions, despite lower mean values; and (2) PSG works as expected for most metrics, *i.e.,* $\tilde{D}_u$ achieves large standardized mean differences from D_u. However, exceptions occur for Orthogonalization and Logit Lens, the same metrics that also suggest unlearning itself was unsuccessful. This indicates that PSG requires the condition that the chosen metric differentiates D_u and D_r in the first place. Intuitively, not being able to do so means the metric performs similarly across the entire training distribution, where an extreme example would be a random metric that decides if a point is unlearned successfully with $50 - 50$ chance. It is impossible to create $\tilde{D}_u$ with a different metric score distribution from D_u in this case. *Therefore, the effectiveness of PSG depends on using metrics that can at least differentiate D_u and D_r.*

7 Discussion

How is the surrogate unlearning dataset related to the unlearning dataset? As aforementioned in § 4.2, PSG ensures the surrogate unlearning dataset is driven from the unlearning dataset by only including sentences with an increase in likelihood when the model is finetuned on the latter. To further confirm this, we analyze D_u and $\tilde{D}_u$ by leveraging topic modeling techniques, under the same setting as in § 6.1. First, by

(a) Negative Preference Optimization (NPO)

(b) Representation Misdirection for Unlearning (RMU)

Figure 6: **Boxplots for metric scores of D_u (*Book X*), $\tilde{D}_u$, and D_r on a smaller model.** Unlike Figure 2, the model architecture used here is Microsoft's Phi-3-mini-4k model with only 3.8 billion parameters, which is significantly less than the other models studied in this work. The results are consistent with Figure 2 for most of the metrics: the boxes representing $\tilde{D}_u$ are consistently different from the ones for $\tilde{D}_u$. However, this is not the case for metrics *Orthogonalization* and *Logit Len*. By comparing the metric scores of D_u and D_r, we suspect this is because unlearning is not successful in the first place, *i.e.*, the metric is performing similarly on all training data (whether it is unlearned or used for retaining), so it is hard to create $\tilde{D}_u$ that is not vastly different from the training distribution and has different metric values at the same time.

clustering the sentences of D_u into topics based on TF-IDF (term frequency–inverse document frequency), we observe 87% sentences of $\tilde{D}_u$ falls into the 5-largest topics of D_u, suggesting similar use of words between the two datasets. We then move to an LLM-based topic modeling technique, BERTopic [16]. Here, we observe 63% sentences of $\tilde{D}_u$ were classified as outliers, *i.e.,* not belonging to any topic clusters of D_u. This may be because that LLM-based topic modeling techniques often rely on embeddings of the underlying LLMs. Recall that PSG maximized embedding distances between the two datasets on θ, so this might transfer to other models like the BERT model used in BERTopic. We then rerun BERTopic

on both D_u and the retain dataset D_r. Despite D_r is much larger in size than D_u, we find 90% sentences of $\tilde{D}_u$ falls into topic clusters dominated by sentences from D_u. This means *$\tilde{D}_u$ is more relevant to D_u than other subsets of the training dataset.*

Could unlearning methods be the cause of the issue? One might argue that the key issue identified by PSG, *i.e.,* unlearning metrics give different scores to sentences in D_u versus sentences learned from D_u, could stem from imperfect unlearning methods. For instance, when applied for copyright or safety, an unlearning method may falsely primarily target exact sentences in D_u, leaving semantically related sentences insufficiently unlearned. This could reproduce our observation: the existence of a dataset related to D_u but with a different distribution of metric values. However, even if this is the case, an ideal metric for copyright- or safety-driven unlearning should report the worst-case performance across both D_u and related data, rather than only the performance on D_u. Otherwise, the metric remains problematic, as it may overstate or understate the success of unlearning. *Thus, imperfect unlearning methods may explain the discrepancy, but they do not absolve the metrics of their responsibility to capture it.*

Could the surrogate unlearning dataset be leveraged during unlearning? Since the surrogate datasets created by PSG consist of sentences that can be learned from D_u, a natural question is whether including them during unlearning would improve unlearning success. The answer depends on the size of $\tilde{D}_u$ and how well it represents the complete distribution of sentences containing knowledge from D_u. If a sufficiently large and representative dataset exists, unlearning it could improve both success and generalizability [30]. However, PSG is designed to falsify metrics, not to generate representative distributions: one counterexample suffices to falsify a metric. As such, unlearning with $\tilde{D}_u$ is analogous to training on a validation set, making evaluation harder without truly improving unlearning[6]. That said, the fine-tuned model θ_f used to construct $\tilde{D}_u$ may still provide useful signals, as it encodes more information about what can be learned from D_u than a few sentences alone. Indeed, similar approaches have been explored in task-vector-based unlearning [22]. Yet relying on a single fine-tuned model is likely insufficient and may lead to failure cases [42]. A more promising direction may be to obtain a distribution of models fine-tuned on D_u and use them collectively to guide unlearning. *Therefore, while $\tilde{D}_u$ itself may not be suitable for unlearning, the fine-tuned models used to construct it could inform better strategies.*

When do unlearning metrics need to be tested against the surrogate dataset? It is worth noting that the surrogate

---

[6]This does not mean including surrogate unlearning datasets in the unlearning dataset can function as an adaptive attack against PSG. As specified in the threat model in § 3.3, the auditor's task is to only evaluate unlearning metrics, so it may use arbitrary unlearning datasets and models. In other words, an adversarial unlearning metric designed against PSG does not have access to the unlearning setting where it would be evaluated.

dataset does not overlap with the unlearning dataset. Therefore, by design, if the goal of unlearning is to remove certain exact data points, the unlearning metrics are not expected to be test against a surrogate dataset—for example, when unlearning is motivated by privacy concerns regarding personally identifiable information. Note that this is not claiming that existing unlearning metrics are perfect in that scenario. In fact, it has been pointed out that the evaluation of privacy-oriented unlearning is also flawed [18]. However, the drawbacks of metrics are different when the goals of unlearning are different, and so are how we should test them—*surrogate datasets are only needed when unlearning beyond sentence-level, e.g., when motivated by copyright or safety.*

What would be the right direction toward best practices for evaluating unlearning? Our results highlight the limitations of existing metrics and suggest properties for improved evaluation. First, metrics must align with the true incentive of unlearning. Measures derived from memorization or membership inference are designed to detect whether a model has memorized or trained on an *exact* data point; naturally, they cannot be expected to generalize to semantically related but unseen data, such as fan fiction of copyrighted content. Second, if the goal is to remove broader knowledge rather than specific data points, the evaluation must also address generalization. This implies two considerations:

1. Evaluation should move beyond individual data points toward sets of examples that all entail the same piece of knowledge, and measure whether that knowledge has been unlearned.

2. Analogous to standard model evaluation, an external test dataset is necessary.

What constitutes a good test dataset remains an open question. The surrogate datasets produced by PSG rely on embedding separation, but this is unlikely to be the only requirement. This also highlights a limitation of PSG itself: a large mean difference between D_u and $\tilde{D}_u$ indicates that a metric is not ideal, but the reverse does not imply that the metric is sound. In other words, PSG is designed to falsify unlearning metrics, not to prove their correctness. *Hence, best practices for evaluating unlearning should move beyond data-point membership, embrace generalization, and develop robust test datasets.* That said, we also recognize the possibility that this work may lead to future findings suggesting it is impossible to faithfully evaluate knowledge-level unlearning, in which case unlearning may not be the best tool for *e.g.,* copyright or safety protection. We hope highlighting this possibility can encourage researchers to revisit fundamental questions on unlearning, such as incentive-dependent best practices for evaluating it, rather than focusing on proposing "better" LLM unlearning methods based on metrics that may not be faithful.

8 Conclusion

We study the problem of evaluating unlearning in generative settings, when the goal is to remove knowledge beyond the data-point level from trained models. We argue that current metrics, which are typically applied only to the unlearning set D_u and thus tailored to D_u, create a false sense of success. In particular, many metrics suggest that data containing knowledge learned from the unlearning dataset has inconsistent unlearning success with respect to the unlearning dataset. To make this gap explicit, we introduced the notion of surrogate unlearning datasets $\tilde{D}_u$, which are derived from (thus semantically related to) but disjoint from D_u, and presented Proximal Surrogate Generation (PSG), a lightweight algorithm for constructing them automatically.

Our empirical results across 18 settings (3 LLMs, 3 datasets, and 2 unlearning algorithms) show that commonly-used unlearning metrics are highly inconsistent between D_u and $\tilde{D}_u$, with $\tilde{D}_u$ frequently appearing less successfully unlearned, according to these metrics. These findings highlight that existing metrics are overfitting to D_u and systematically fail to evaluate unlearning success based on the removal of higher-order knowledge, which is more tied to the incentive of knowledge-based (*e.g.,* copyrighted contents removal or unsafe outputs mitigation) unlearning. Looking ahead, we hope our work can encourage future unlearning metrics to account for generalization of unlearning (*e.g.,* by the use of a pseudo-validation dataset) and tailor to the goal of unlearning: in particular, when unlearning is motivated by knowledge removal, the metrics can be designed to move beyond exact-data deletion and toward capturing the true extent of knowledge removal in LLMs.

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. *arXiv e-prints*, page arXiv:2404.14219, April 2024.

[2] George-Octavian Barbulescu and Peter Triantafillou. To each (textual sequence) its own: Improving memorized-data unlearning in large language models. *arXiv preprint arXiv:2405.03097*, 2024.

[3] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. *CoRR*, abs/1903.04561, 2019.

[4] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning, 2020.

[5] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480, 2015.

[6] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association, August 2021.

[7] Somnath Basu Roy Chowdhury, Krzysztof Choromanski, Arijit Sehanobish, Avinava Dubey, and Snigdha Chaturvedi. Towards scalable exact machine unlearning using parameter-efficient fine-tuning, 2025.

[8] A Feder Cooper, Christopher A Choquette-Choo, Miranda Bogen, Matthew Jagielski, Katja Filippova, Ken Ziyu Liu, Alexandra Chouldechova, Jamie Hayes, Yangsibo Huang, Niloofar Mireshghallah, et al. Machine unlearning doesn't do what you think: Lessons for generative ai policy, research, and practice. *arXiv preprint arXiv:2412.06966*, 2024.

[9] Guangyao Dou, Zheyuan Liu, Qing Lyu, Kaize Ding, and Eric Wong. Avoiding copyright infringement via large language model unlearning. *arXiv preprint arXiv:2406.10952*, 2024.

[10] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*, 2017.

[11] Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms, 2023.

[12] Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning, 2025.

[13] Jean-loup Gailly and Mark Adler. zlib compression library.

[14] Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgeting in gradient-based neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[15] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models, 2024.

[16] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[17] Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi Malvajerdi, and Christopher Waites. Adaptive machine unlearning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[18] Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. Inexact Unlearning Needs More Careful Evaluations to Avoid a False Sense of Privacy . In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 497–519, Los Alamitos, CA, USA, April 2025. IEEE Computer Society.

[19] Jing Huang, Diyi Yang, and Christopher Potts. Demystifying verbatim memorization in large language models. *arXiv preprint arXiv:2407.17817*, 2024.

[20] Zhehao Huang, Xinwen Cheng, JingHao Zheng, Haoran Wang, Zhengbao He, Tao Li, and Xiaolin Huang. Unified gradient-based machine unlearning with remain geometry enhancement. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[21] Dang Huu-Tien, Trung-Tin Pham, Hoang Thanh-Tung, and Naoya Inoue. On effects of steering latent representation for large language model unlearning, 2025.

[22] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

[23] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models, 2022.

[24] Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana R Kompella, Sijia Liu, and Shiyu Chang. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *Advances in Neural Information Processing Systems*, 37:12581–12611, 2024.

[25] Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. Soul: Unlocking the power of second-order optimization for llm unlearning. *arXiv preprint arXiv:2404.18239*, 2024.

[26] Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7403–7412, Singapore, December 2023. Association for Computational Linguistics.

[27] Aly Kassem, Omar Mahmoud, and Sherif Saad. Preserving privacy through dememorization: An unlearning technique for mitigating memorization risks in language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4360–4379, Singapore, December 2023. Association for Computational Linguistics.

[28] Nathaniel Li, Alexander Pan, Anjali Gopal, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning, 2024.

[29] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pages 1–14, 2025.

[30] Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Machine unlearning in generative ai: A survey. *arXiv preprint arXiv:2407.20516*, 2024.

[31] Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. Quark: Controllable text generation with reinforced unlearning, 2022.

[32] Jakub Łucki, Boyi Wei, Yangsibo Huang, Peter Henderson, Florian Tramèr, and Javier Rando. An adversarial perspective on machine unlearning for AI safety. *Transactions on Machine Learning Research*, 2025.

[33] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms, 2024.

[34] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

[35] Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[36] Vaidehi Patil, Peter Hase, and Mohit Bansal. Can sensitive information be deleted from llms? objectives for defending against extraction attacks, 2023.

[37] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[38] Jie Ren, Zhenwei Dai, Xianfeng Tang, Hui Liu, Jingying Zeng, Zhen Li, Rahul Goutam, Suhang Wang, Yue Xing, and Qi He. A general framework to enhance fine-tuning-based llm unlearning. *arXiv preprint arXiv:2502.17823*, 2025.

[39] Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary Chase Lipton, and J Zico Kolter. Rethinking LLM memorization through the lens of adversarial compression. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[40] William F. Shen, Xinchi Qiu, Meghdad Kurmanji, Alex Iacob, Lorenzo Sani, Yihong Chen, Nicola Cancedda, and Nicholas D. Lane. Lunar: Llm unlearning via neural activation redirection, 2025.

[41] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[42] Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models, 2024.

[43] Pratiksha Thaker, Shengyuan Hu, Neil Kale, Yash Maurya, Zhiwei Steven Wu, and Virginia Smith. Position: LLM unlearning benchmarks are weak measures of progress. In *IEEE Conference on Secure and Trustworthy Machine Learning, SaTML 2025, Copenhagen, Denmark, April 9-11, 2025*, pages 520–533. IEEE, 2025.

[44] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023.

[45] Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing, Cham, 1st edition, 2017.

[46] Qizhou Wang, Bo Han, Puning Yang, Jianing Zhu, Tongliang Liu, and Masashi Sugiyama. Towards effective evaluations and comparisons for llm unlearning methods. In *The Thirteenth International Conference on Learning Representations*, 2025.

[47] Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q Weinberger. Rethinking llm unlearning objectives: A gradient perspective and go beyond. *arXiv preprint arXiv:2502.19301*, 2025.

[48] Wenyu Wang, Mengqi Zhang, Xiaotian Ye, Zhaochun Ren, Zhumin Chen, and Pengjie Ren. Uipe: Enhancing llm unlearning by removing knowledge related to forgetting targets. *arXiv preprint arXiv:2503.04693*, 2025.

[49] Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Parag Shah, Yujia Bao, Yang Liu, and Wei Wei. Llm unlearning via loss adjustment with only forget data, 2024.

[50] Jiaxin Wen, Pei Ke, Hao Sun, Zhexin Zhang, Chengfei Li, Jinfeng Bai, and Minlie Huang. Unveiling the implicit toxicity in large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1322–1338, Singapore, December 2023. Association for Computational Linguistics.

[51] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Lia, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024.

[52] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.

[53] Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. Min-k%++: Improved baseline for pre-training data detection from large language models. In *The Thirteenth*

International Conference on Learning Representations, 2025.

[54] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning, 2024.

[55] Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.

Appendix

A Experimental Setup

A.1 Datasets and Retain/Unlearning Mapping

We organize experiments into three tasks: *Novel*, *WMDP*, and *Toxic*. Sizes and dataset identifiers are summarized in Table 2. Below we state, for each task, the precise mapping from the unlearning dataset (also known as forget dataset) to its corresponding retain dataset(s), consistent with the main text and tables.

Novel task. We fine-tune on 12 novels and their associated fan fiction. For unlearning, the **unlearning** dataset is Book X. The **retain** dataset is the union of the other 11 novels, as listed in Table 2.

WMDP task. The **unlearning** dataset is *WMDP-bio* [28]. The **retain** dataset is *Salesforce/wikitext* (specifically `wikitext-2-raw-v1`) [34], as listed in Table 2.

Toxic task. We fine-tune on the union of `google/civil_comments` subsets with (i) toxicity$= 0$ and (ii) toxicity$\geq$0.5 [3]. For unlearning, the **unlearning** dataset is the toxicity$\geq$0.5 subset; the **retain** dataset is the toxicity$= 0$ subset, as listed in Table 2.

Table 2: Dataset sizes for corpora used in the WMDP, Toxic, and Novel tasks.

Dataset	Comment	Size
WMDP-bio [28]	bio-forget-corpus	712.96 MB
Salesforce/wikitext [34]	wikitext-2-raw-v1	12.77 MB
google/civil_comments [3]	toxicity $= 0$	348.42 MB
google/civil_comments	toxicity ≥ 0.5	38.38 MB
Novel task (fine-tune)	All novels + fan fiction	81.90 MB
Novel task (unlearning)	One novel (Book X)	0.63 MB
Novel task (retain)	Other 11 novels	33.85 MB

A.2 Models and Training Environment

We conduct experiments with *Meta-Llama-3-8B* [15], *Qwen2.5-7B* [51], and *Zephyr-7B-beta* [44]. For unlearning, we employ *Negative Preference Optimization (NPO)* [54] and *Removal via Model Update (RMU)* as instantiated in the WMDP framework [28].

Hardware. All runs used four NVIDIA A100-SXM4-80GB GPUs (80 GB each). The host has two AMD EPYC 7643 48-core processors (96 physical cores / 192 threads). The total system RAM was not captured in our environment logs. The OS is Ubuntu 22.04.5 LTS (64-bit).

Software. We used Python 3.11.9 and PyTorch v2.6.0 with CUDA Toolkit 12.1. The NVIDIA driver is 535.161.08 (supports up to CUDA 12.2).

B Is line 7 of Proximal Surrogate Generation (PSG) jailbreaking?

Although our method adopts a GCG-like optimization, it is fundamentally different from jailbreaking. In jailbreaking, free tokens are optimized to coerce the model into producing a specific harmful or restricted continuation. In contrast, our objective does not prescribe any target output. Instead, the free tokens are optimized solely to maximize separation from D_u in the embedding space. The outcome is not unsafe generations, but natural sentences that remain semantically related to D_u while stress-testing unlearning metrics. The similarity to GCG is thus purely structural, not adversarial in either intent or use. In short, PSG borrows the mechanics of GCG but serves a fundamentally different, non-adversarial purpose.
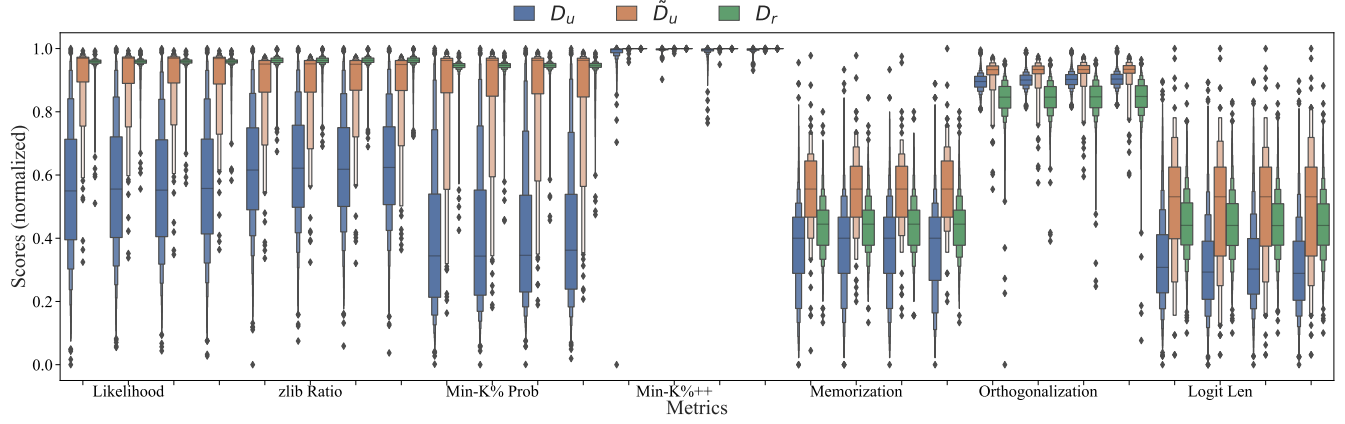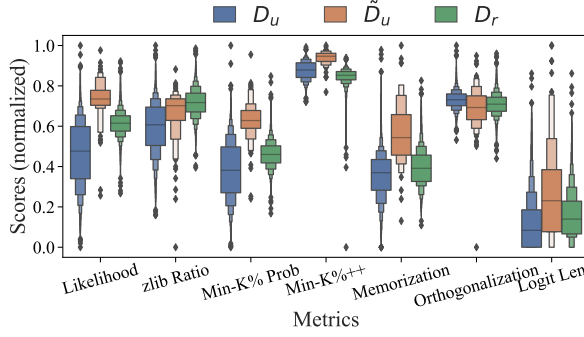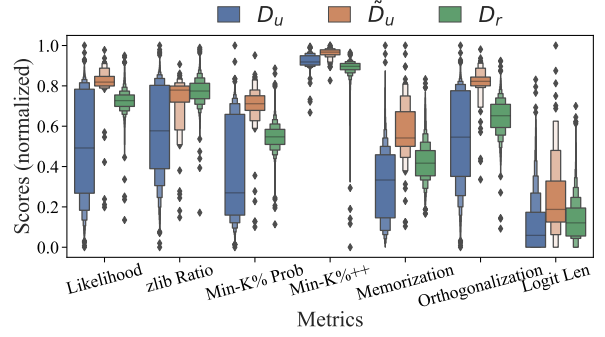
C   Additional Experiment Results

Figure 7: Boxplots for metric scores of D_u (*Book X*), $\tilde{D}_u$, and D_r where the unlearning method is NPO and the model architecture is Llama-3. With the same model before unlearning, we perform unlearning 4 times with identical hyperparameters but different random seeds. Therefore, there are 4 boxes for each of D_u, $\tilde{D}_u$, and D_r for every metric. Note that, besides D_u and D_r, the $\tilde{D}_u$ is also identical across the 4 runs of unlearning, since Proximal Surrogate Generation (PSG) is not impacted by the unlearned model. One can see that PSG performs consistently across different random seeds.
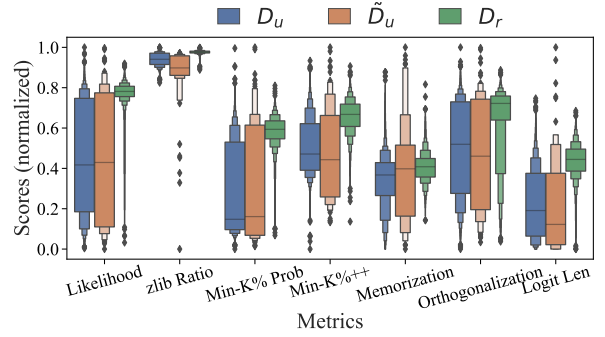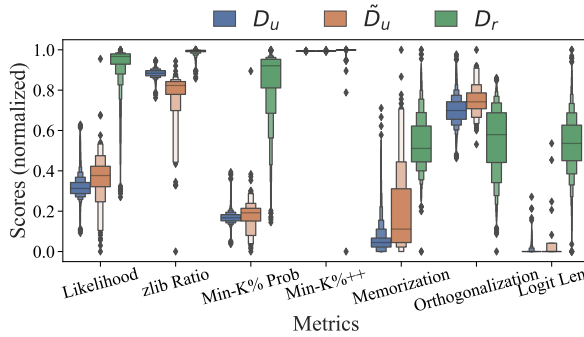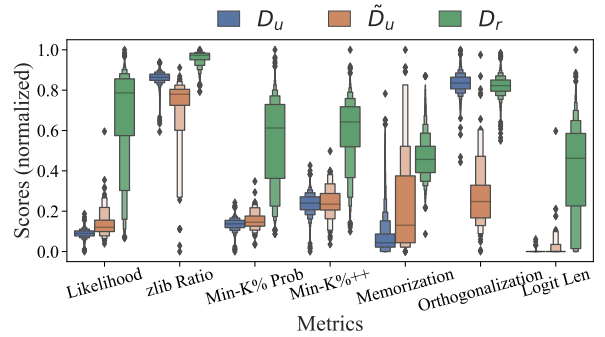
(a) Negative Preference Optimization (NPO)

(b) Representation Misdirection for Unlearning (RMU)

Figure 8: This is a reproduction of Figure 2 with identical settings, except the model architecture is Qwen2.5.

(a) Negative Preference Optimization (NPO)

(b) Representation Misdirection for Unlearning (RMU)

Figure 9: This is a reproduction of Figure 2 with identical settings, except the model architecture is Zephyr.
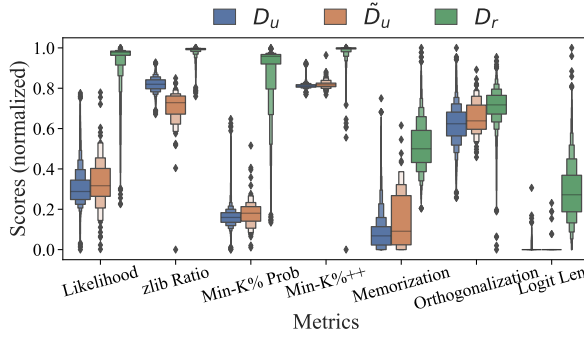

(a) Negative Preference Optimization (NPO)
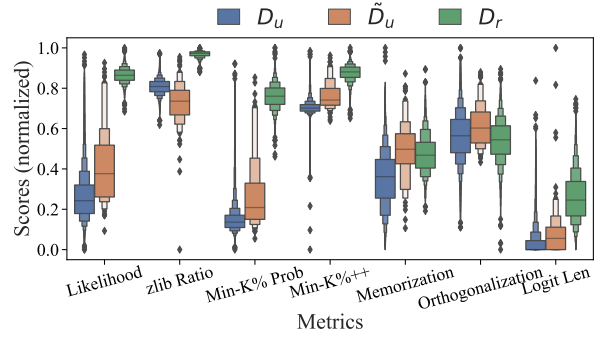
(b) Representation Misdirection for Unlearning (RMU)

Figure 10: This is a reproduction of Figure 2 with identical settings, except the dataset is Civil Comments. Note that here, for computational efficiency, the metric values are computed for 5% of D_u (randomly sampled), which contain about 13 thousand sentences.

(a) Negative Preference Optimization (NPO)

(b) Representation Misdirection for Unlearning (RMU)

Figure 11: This is a reproduction of Figure 2 with identical settings, except the dataset is Civil Comments, and the model architecture is Qwen2.5.
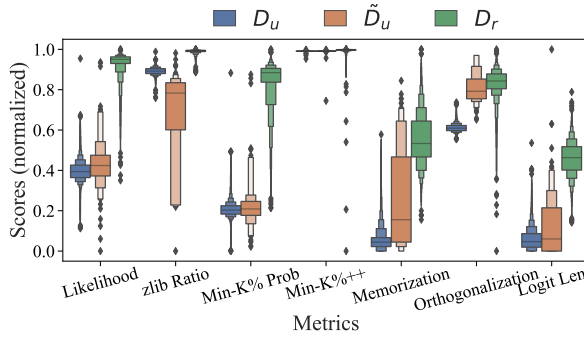

(a) Negative Preference Optimization (NPO)

(b) Representation Misdirection for Unlearning (RMU)

Figure 12: This is a reproduction of Figure 2 with identical settings, except the dataset is Civil Comments, and the model architecture is Zephyr.

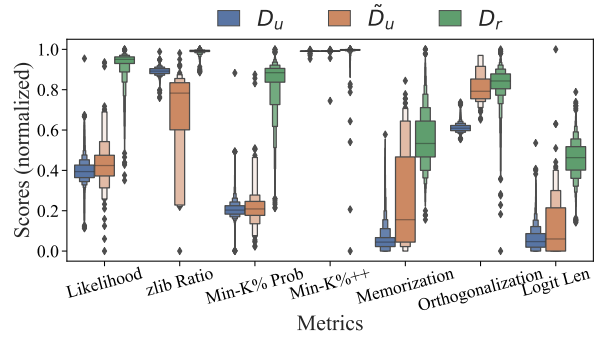(a) Negative Preference Optimization (NPO)

(b) Representation Misdirection for Unlearning (RMU)

Figure 13: This is a reproduction of Figure 2 with identical settings, except the dataset is WMDP-bio. Note that here, for computational efficiency, the metric values are computed for 1% of D_u (randomly sampled), which contain about 18 thousand sentences.

(a) Negative Preference Optimization (NPO)

(b) Representation Misdirection for Unlearning (RMU)

Figure 14: This is a reproduction of Figure 2 with identical settings, except the dataset is WMDP-bio, and the model architecture is Qwen2.5.

(a) Negative Preference Optimization (NPO)

(b) Representation Misdirection for Unlearning (RMU)

Figure 15: This is a reproduction of Figure 2 with identical settings, except the dataset is WMDP-bio, and the model architecture is Zephyr.
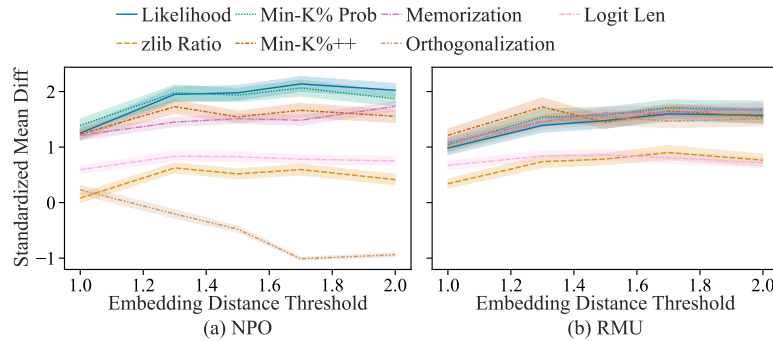

(a) NPO

(b) RMU

Figure 16: This is a reproduction of Figure 4 with identical settings, except the model architecture is Qwen2.5. Similar trends can be observed.
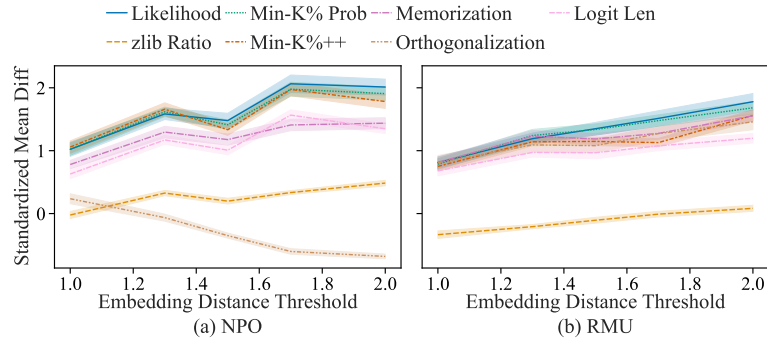
Figure 17: This is a reproduction of Figure 4 with identical settings, except the model architecture is Zephyr. Similar trends can be observed.