WaterDrum: Watermarking for Data-centric Unlearning Metric

Anonymous Authors[1]

Abstract

Large language model (LLM) unlearning is critical in real-world applications where it is necessary to efficiently remove the influence of private, copyrighted, or harmful data from some users. However, existing utility-centric unlearning metrics (based on model utility) may fail to accurately evaluate the extent of unlearning in realistic settings such as when (a) the forget and retain sets have semantically similar content, (b) retraining the model from scratch on the retain set is impractical, and/or (c) the model owner can improve the unlearning metric without directly performing unlearning on the LLM. This paper presents the first data-centric unlearning metric for LLMs called `WaterDrum` that exploits robust text watermarking to overcome these limitations. We introduce new benchmark datasets for LLM unlearning that contain varying levels of similar data points and can be used with `WaterDrum` to rigorously evaluate unlearning algorithms.

1. Introduction

The capabilities of large language models (LLMs) have drastically improved in recent years, prompting increased efforts to deploy LLMs in real-world applications. However, accompanying this push for practical LLM deployment are growing concerns around data issues regarding LLMs that may threaten to derail such developments, especially since LLMs typically require large amounts of training data. Data owners have raised *intellectual property (IP) infringement* concerns: For example, the New York Times has sued OpenAI over its LLM's use of their copyrighted work (Grynbaum and Mac, 2023). Many jurisdictions are also paying increased scrutiny over *data privacy* concerns, e.g., with regulations such as the General Data Protection Regulation (GDPR, 2016) and the California Consumer Privacy Act (CCPA, 2018) mandating the "right to be forgotten" that allow users to request the erasure of their data from the trained models. Furthermore, it is also not uncommon for public data to become outdated or to be found erroneous/harmful, e.g., the retraction of public scientific papers with fabricated data (Hu et al., 2024).

These data concerns have sparked considerable research efforts on LLM unlearning algorithms, which aim to efficiently remove the influence of a subset of the model's original training data (called the *forget set*) while avoiding the prohibitively expensive alternative of retraining the model from scratch on the *retain set*. However, due to the size and complexity of LLMs, existing unlearning algorithms cannot yet achieve perfect unlearning like retraining: They may not fully remove the influence of all data in the forget set, and may also inadvertently remove the influence of data in the retain set that should be preserved (Maini et al., 2024; Shi et al., 2025). This raises a natural question: *How can we measure the extent to which these algorithms have unlearned a given set of data?* Existing works have largely proposed utility-centric unlearning metrics that evaluate unlearning based on model utility (performance) indicators, like the perplexity or accuracy on downstream tasks. After unlearning, the model utility indicators related to the forget set are expected to worsen. We provide an overview of existing utility, membership inference attack, and image and classification watermarking based unlearning metrics in App. A.1 and position our work with respect to other LLM unlearning evaluation works in App. A.2.

However, *are the utility-centric metrics effective in the face of practical challenges with real-world datasets?* In real-life settings, it is (a) common for the forget and retain sets to have semantically similar content, (b) typical to be prohibitively expensive to retrain an LLM, and (c) possible that an LLM owner might attempt to improve the metric without directly performing LLM unlearning to reduce cost. In App. H, we will show that utility-centric metrics fall short and we have identified two reasons. Expecting worse utility on the forget set after unlearning ignores the ability of the LLMs to generalize from the retain set (Liu et al., 2024). In addition, to evaluate the success of unlearning, these metrics require referencing the retrained LLM (on the retain set) which cannot be obtained in practice.

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

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

Table 1: Comparison of unlearning metrics based on the proposed desiderata (Sec. 2). We enforce **D3**, so the metrics cannot rely on the retrained model. **D1** and **D2** consider the setting with an honest model owner and no similar data.

	D1	D2	D4
ROUGE (Maini et al., 2024)	✓	✗	✗
Truth Ratio (Maini et al., 2024)	✓	✗	✗
KnowMem (Shi et al., 2025)	✓	✗	✗
MIA (Shi et al., 2024)	✗	✗	✗
`WaterDrum` (ours)	✓	✓	✓

In this work, we **(a)** define clear desiderata that an effective and practical unlearning metric should satisfy (Sec. 2), **(b)** propose a novel *data-centric* approach to evaluating the success of LLM unlearning instead, which we call **Water**marking for **D**ata-cent**r**ic **U**nlearning **M**etric (`WaterDrum`) that satisfies these desiderata, and **(c)** propose a new benchmark dataset `WaterDrum-Ax` that includes data from multiple parties and contains duplicates with varying degrees of similarity (App. F). `WaterDrum` is based on a robust text watermarking framework `Waterfall` (Lau et al., 2024) that is capable of verifying multiple data owners' watermarks in LLM outputs when the LLM is trained on their watermarked data (Sec. 3). Our key insight is that using watermarked data creates a clear counterfactual — a model that is not trained on watermarked data would not contain the watermark signal.

2. Problem Formulation and Desiderata

We consider the setting of a collection $\mathcal{T}$ of data owners where each data owner i has a dataset $\mathcal{D}_i$. These datasets may contain similar data instances (e.g., news articles on the same event from different news agencies or paper abstracts from the same arXiv category but different authors, as illustrated in App. K.3). The model owner aggregates their data $\mathcal{D}_\mathcal{T} \coloneqq \bigcup_{i\in\mathcal{T}} \mathcal{D}_i$ for training an LLM $\varphi_\mathcal{T}$ to be deployed as a service. We consider the unlearning scenario where a subset $\mathcal{F} \subseteq \mathcal{T}$ of data owners requests to remove the influence of their to-be-erased data $\mathcal{D}_\mathcal{F} \coloneqq \bigcup_{i\in\mathcal{F}} \mathcal{D}_i$ (called the *forget set*) from the LLM due to concerns about privacy, IP protection, or erroneous content.

Ideally, the model owner would retrain a new model $\varphi_\mathcal{R}$ on the remaining set of data $\mathcal{D}_{\mathcal{R}\coloneqq\mathcal{T}\setminus\mathcal{F}}$ (called the *retain set*) to comply with these unlearning requests. However, full retraining is impractical in reality due to the prohibitive computational cost, especially when $\mathcal{D}_\mathcal{R}$ is large. Instead, the model owner would resort to using some *unlearning algorithm*, which modifies the original model $\varphi_\mathcal{T}$ based on $\mathcal{D}_\mathcal{F}$ to an *unlearned model* $\widetilde{\varphi}$ that approximates $\varphi_\mathcal{R}$. Such an unlearned model may not have perfectly unlearned the forget set, so it can be intuitively viewed as retaining the influence of some (possibly unknown) subset of the forget set $\mathcal{D}_\CIRCLE \subseteq \mathcal{D}_\mathcal{F}$ and hence still be effectively influenced by its **approximate retain set** $\mathcal{D}_\mathcal{R} \bigcup \mathcal{D}_\CIRCLE$. Note that $\mathcal{D}_\CIRCLE$ might not correspond exactly to the union of $\mathcal{D}_i$'s over some subset of data owners in $\mathcal{F}$ and can possibly include only a subset of data points from each $\mathcal{D}_i$. The best unlearned models should have $|\mathcal{D}_\CIRCLE|$ and its influence to be as small as possible.

In most practical scenarios, data owners have **only query access to the model**. Let the query function q denote a mapping from each given data point $d_\bullet$ or dataset $\mathcal{D}_\bullet$ to a corresponding text query $q(d_\bullet)$ or set $q(\mathcal{D}_\bullet)$ of text queries. For example, q can be a function that structures $d_\bullet$ into an appropriate prompt format $q(d_\bullet)$ to query an LLM for Q&A or completion tasks. To ease notation, we abbreviate $q(\mathcal{D}_\bullet)$ as $q_\bullet$; for example, q_i and $q_\mathcal{F}$ denote the queries formed using $\mathcal{D}_i$ and $\mathcal{D}_\mathcal{F}$, respectively.

To analyze whether the model owner has unlearned their dataset $\mathcal{D}_i$, the data owner i can rely on some LLM output, such as $\varphi_\bullet(q(d_i))$ or $\varphi_\bullet(q_i)$, to compute an *unlearning metric* M that quantifies the extent to which their data remains present in the output. Specifically, we define an unlearning metric M where $M(\varphi_\bullet(q(d_i)), i)$ and $M(\varphi_\bullet(q_i), i)$, respectively, measure the influence of i's data $\mathcal{D}_i$ (i.e., second input to M) detectable in the output of LLM $\varphi_\bullet$ to queries $q(d_i)$ or q_i. Additionally, for ease of notation, we also use M to measure the influence of data from a set of owners; for example, $M(\varphi_\mathcal{R}(q_\mathcal{F}), \mathcal{F})$ measures the influence of the forget set $\mathcal{D}_\mathcal{F}$ detectable in the retrained LLM $\varphi_\mathcal{R}$'s output. The metric M should satisfy the following non-exhaustive desiderata.

2.1. Effectiveness

First, the metric must effectively measure the extent to which an unlearning algorithm has not unlearned the forget set (so, the resulting unlearned LLM $\widetilde{\varphi}$ would still be influenced by its unknown approximate retain set, as discussed in Sec. 2). To achieve this, we will now define effectiveness desiderata that utilize LLMs retrained on the retain set (and varying known subsets of the forget set) as retraining is a perfect unlearning algorithm:[1]

D1 Separability. The metric should detect/classify whether an owner's data still has influence on an unlearned LLM. Specifically, when evaluating the retrained LLM $\varphi_\mathcal{R}$ (i.e., achieved by perfect unlearning), the metric should, with high probability, **give higher values when measured on the output to queries formed by the retain set** $\mathcal{D}_\mathcal{R}$ (which influences $\varphi_\mathcal{R}$) **than queries formed by the forget set** $\mathcal{D}_\mathcal{F}$ (which does not). That is, for any randomly selected data points $d_r \in \mathcal{D}_r \subseteq \mathcal{D}_\mathcal{R}$ from owner r and $d_f \in \mathcal{D}_f \subseteq \mathcal{D}_\mathcal{F}$

[1]The retrained LLMs are only used to justify our effectiveness desiderata for evaluating the unlearning metrics. In practice, the metrics should be used to evaluate imperfect unlearning algorithms without the retrained LLMs, as discussed in **D3**(a).

from owner f, the probability

$$\mathbb{P}\left[M(\varphi_{\mathcal{R}}(q(d_r)), r) > M(\varphi_{\mathcal{R}}(q(d_f)), f)\right] \approx 1\,. \quad (1)$$

Separability, which is defined by the left-hand side of Eq. (1) (or, equivalently, AUROC), implies that some threshold κ exists such that for any data point $d_i \in \mathcal{D}_i \subseteq \mathcal{D}_{\mathcal{T}}$ from owner i, a large value $M(\varphi_{\mathcal{R}}(q(d_i)), i) > \kappa$ indicates that d_i is likely to be in the retain set $\mathcal{D}_{\mathcal{R}}$; varying κ yields the ROC curve. Similarly, when considering an unlearned LLM $\widetilde{\varphi}$, a large value $M(\widetilde{\varphi}(q(d_i)), i)$ indicates that d_i is likely to be in the approximate retain set (Sec. 2). In other words, the metric should serve as a good classifier for whether an owner's data still influences the LLM and is hence in the approximate retain set: A higher AUROC indicates a better separability of data that influences the LLM vs. not (Fawcett, 2006). Further discussion is given in App. D.1.

D2 Calibration. In Sec. 1, we highlighted that existing unlearning algorithms cannot yet achieve perfect unlearning. Thus, our unlearning metric should be **calibrated to the extent of imperfect unlearning**. For example, we can simulate different extents of imperfect unlearning by retraining with different sizes of subsets of the forget set. Specifically, the metric (in expectation) should be proportional to the size k of the random subset $\mathcal{D}_{\textcircled{\scriptsize}}$ of the forget set that is used to retrain the LLM $\widehat{\varphi}$:

$$\mathbb{E}_{\mathcal{D}_{\textcircled{\scriptsize}} \subseteq \mathcal{D}_{\mathcal{F}} : |\mathcal{D}_{\textcircled{\scriptsize}}| = k}\left[M(\widehat{\varphi}(q_{\mathcal{F}}), \mathcal{F})\right] \;\propto\; k/|\mathcal{D}_{\mathcal{F}}| \quad (2)$$

where $\mathcal{D}_{\textcircled{\scriptsize}}$ is defined in a similar way as $\mathcal{D}_{\textcircled{\scriptsize}}$ in Sec. 2 except that it is known. Eq. (2) implies that a perfectly unlearned LLM like $\varphi_{\mathcal{R}}$ should have $M(\varphi_{\mathcal{R}}(q_{\mathcal{F}}), \mathcal{F}) = 0$ since $k = 0$. So, when evaluating unlearning algorithms, we identify successful perfect unlearning of the forget set by looking for $M(\widetilde{\varphi}(q_{\mathcal{F}}), \mathcal{F}) \approx 0$. In addition, the metric's value can be intuitively interpreted as the extent to which the forget set has not been unlearned. This enables the unlearning metric to go beyond being just a binary indicator of whether an entire forget set is unlearned to a meaningful continuous score of unlearning. Further discussion is given in App. D.2.

2.2. Practicality

A viable metric for deployment must satisfy the following additional feasibility and robustness desiderata that account for challenges faced in common real-life scenarios:

D3 Feasibility. (a) When the metric is used to evaluate an unlearning algorithm and produce $M(\widetilde{\varphi}(q_i), i)$ on the unlearned LLM $\widetilde{\varphi}$, it **should not require the retrained model** $\varphi_{\mathcal{R}}$. The premise of unlearning is that retraining the model on the retain set is prohibitively expensive. Hence, metrics cannot depend on $\varphi_{\mathcal{R}}$ in practice. However, as we will see in Secs. F.1 and H.3, many existing metrics cannot satisfy **D2** without access to $\varphi_{\mathcal{R}}$, which limits their practical use. (b) To enable data owners with only query access to the LLM to evaluate unlearning, the metric **should only depend on the queried output** instead of full access to the weights or token probabilities of the unlearned model $\widetilde{\varphi}$.

D4 Robustness to similar data. The effectiveness desiderata **D1**-**D2** should hold for any $\mathcal{D}_{\mathcal{R}}$ and $\mathcal{D}_{\mathcal{F}}$, including typical scenarios where $\mathcal{D}_{\mathcal{R}}$ and $\mathcal{D}_{\mathcal{F}}$ have similar data points. We further discuss this desideratum in App. D.3.

3. Methodology

3.1. Challenges for utility-centric unlearning metrics

Utility-centric unlearning metrics have evaluated the effectiveness of unlearning based on model utility (performance) indicators, such as verbatim memorization, perplexity, and accuracy on downstream tasks. Performance indicators P have compared the unlearned LLM $\widetilde{\varphi}$'s output to queries (e.g., $\widetilde{\varphi}(q_{\mathcal{F}})$ on the forget set) to the original data (e.g., $\mathcal{D}_{\mathcal{F}}$). We describe several types of utility-centric unlearning metrics in App. G.2.

However, such performance indicators P do not meet our required desiderata for the metric M (Sec. 2). First, **D3**(a) does not allow retraining the LLM. Without retraining, the reference value $P(\varphi_{\mathcal{R}}(q_{\mathcal{F}}), \mathcal{D}_{\mathcal{F}})$ of the perfectly unlearned LLM (i.e., retrained LLM $\varphi_{\mathcal{R}}$) cannot be determined and thus cannot be used to offset the metric to produce a value close to 0 when the forget set $\mathcal{D}_{\mathcal{F}}$ is perfectly unlearned (e.g., it is not possible to define and compute M as $P(\widetilde{\varphi}(q_{\mathcal{F}}), \mathcal{D}_{\mathcal{F}}) - P(\varphi_{\mathcal{R}}(q_{\mathcal{F}}), \mathcal{D}_{\mathcal{F}})$). Thus, without retraining, P does not satisfy **D2**, making it difficult to identify successful unlearning of the forget set. Next, when there are similar data present in the forget and retain sets (**D4**), we observe that any unlearned LLM $\widetilde{\varphi}$ (e.g., the retrained LLM $\varphi_{\mathcal{R}}$) tends to produce similar outputs to queries on both sets, that is, $\widetilde{\varphi}(q_{\mathcal{F}}) \simeq \widetilde{\varphi}(q_{\mathcal{R}})$, as empirically verified in App. I.2. As the performance indicators largely depend on direct comparisons with the LLM outputs, their corresponding values will also be similar, i.e., $P(\widetilde{\varphi}(q_{\mathcal{F}}), \mathcal{D}_{\mathcal{F}}) \approx P(\widetilde{\varphi}(q_{\mathcal{R}}), \mathcal{D}_{\mathcal{R}})$. We will show in App. H that this leads to utility-centric metrics failing to satisfy **D1** when the data from the forget and retain sets are highly or moderately similar. The failure arises because expecting poor predictions on the forget set and a low $P(\widetilde{\varphi}(q_{\mathcal{F}}), \mathcal{D}_{\mathcal{F}})$ overlooks the generalization capability of LLMs (Liu et al., 2024). Table 1 presents a comparison of our `WaterDrum` and existing metrics based on the desiderata in Sec. 2. In App. A.1, we provide further details on utility-centric and other unlearning metrics.

3.2. Watermarking as unlearning metric

To overcome the challenges described above and satisfy the desiderata in Sec. 2, we propose to adopt a novel *data-centric* approach to evaluating the success of

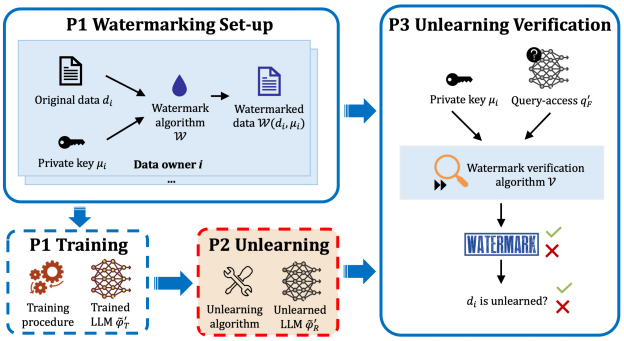


Figure 1: Overview of the watermarking, training, unlearning, and verification process in `WaterDrum`.

unlearning instead. Instead of relying on utility-centric metrics that indirectly infer unlearning via model performance, we **directly track the presence of data by actively embedding data-specific signals detectable in the LLM outputs that are designed to be orthogonal to its performance**. In App. A.1, we highlight how `WaterDrum` differs from existing watermarking-based metrics for image classification tasks. In App. B, we outline desiderata required by a watermarking framework (and its verification operator) to meet our unlearning metric desiderata in Sec. 2.

3.3. Overview of `WaterDrum` and Experimental results

To satisfy the watermarking desiderata presented in App. B, we propose `WaterDrum`, an unlearning metric built on top of our adaptation of the scalable and robust `Waterfall` framework (Lau et al., 2024) which can successfully verify multiple owners' watermarks in LLM outputs when the LLM has been trained on their watermarked text.

Specifically, we adopt the watermarking $\mathcal{W}(\cdot, \mu)$ and verification $\mathcal{V}(\cdot, \mu)$ operators as defined in `Waterfall`. We can then define the `WaterDrum` metric on datasets as

$$M'(\varphi_\bullet(q'_i), \cdot) := |\mathcal{D}'_i|^{-1} \sum_{d'_i \in \mathcal{D}'_i} M'(\varphi_\bullet(q(d'_i)), \cdot). \quad (3)$$

`Waterfall`'s watermarking and verification approaches satisfy the watermarking desiderata **W0**, **W1**(a) and **W2**, as elaborated and demonstrated in (Lau et al., 2024). We empirically verified that the `Waterfall` method satisfies **W0** in App. I.1 and **W1**(b) on calibration in App. H.3. The rest of the watermarking process desiderata can be satisfied by properly designing the unlearning and evaluation process, which we illustrate in Fig. 1 and present below:

P1 Watermarking setup. Each data owner i first watermarks their data $\mathcal{D}_i$ with a unique private key μ_i to generate a watermarked dataset $\mathcal{D}'_i := \{d'_i = \mathcal{W}(d_i, \mu_i) \mid d_i \in \mathcal{D}_i\}$, before the model owner aggregates their watermarked data $\mathcal{D}'_\mathcal{T} := \bigcup_{i \in \mathcal{T}} \mathcal{D}'_i$, trains a model $\varphi'_\mathcal{T}$ on it, and offer to clients (including data owners) query access to the trained model.

P2 Unlearning. A subset of data owners $\mathcal{F}$ requests that their data $\mathcal{D}'_\mathcal{F} := \bigcup_{i \in \mathcal{F}} \mathcal{D}'_i$ be erased from the model $\varphi'_\mathcal{T}$. The model owner will claim to have done the unlearning, and offer query access to a new model $\widetilde{\varphi}'$.

P3 Unlearning verification. The verification operator takes the role of the uncertainty metric in `WaterDrum`, as per Eq. (3). In most cases, each data owner f in $\mathcal{F}$ can query the unlearned model $\widetilde{\varphi}'$ with queries q'_f based on $\mathcal{D}'_f$ and apply the verification operator $\mathcal{V}(\widetilde{\varphi}(q'_f), \mu_f)$ to measure the extent that their data has been unlearned. More challenging situation of queries, such as under a threat model, is described later in App. E.

Note that `WaterDrum` in Eq. (3) applied during **P3** only requires query access to the model, hence satisfying **W3**. Watermarking desideratum **W4** is also satisfied by the setup in **P1** and the fact that the model owner never requires the data owners' keys, including in **P2**. In App. K.1, we explain why the process is practical and discuss deployment details.

To effectively evaluate `WaterDrum` and compare with other baseline unlearning metrics, we explain the limitations of existing unlearning benchmark datasets and introduce a new benchmark dataset `WaterDrum-Ax` in App. F.

Experiments. We empirically verify whether `WaterDrum` and the baseline unlearning metrics satisfy the proposed desiderata – see App. H for details). Below, we extract two key results showing how `WaterDrum` is the only metric that consistently satisfy **D1** (Table 2), and **D2** (Table 3), under varying levels of similar data across $\mathcal{D}_\mathcal{R}$ and $\mathcal{D}_\mathcal{F}$ (**D4**) and enforcing **D3** (i.e., no access to retrained model).

Table 2: AUROC ($\pm$ across 3 seeds) of metrics for different levels of similarity for the `WaterDrum-Ax` dataset.

Similarity	ROUGE	KnowMem	`WaterDrum`
Exact Dup.	0.334±0.005	0.492±0.005	**0.957±0.008**
Sem. Dup.	**0.960±0.002**	0.450±0.007	**0.963±0.001**
No Dup.	**0.974±0.001**	0.491±0.008	**0.965±0.002**

Table 3: R^2 of the best fit line (dotted in Fig. 6) for metrics under different levels of similarity on the `WaterDrum-Ax` dataset. `WaterDrum` is very well linearly calibrated across the settings, with the highest R^2 value.

Similarity	ROUGE	KnowMem	MIA	`WaterDrum`
Exact Dup.	-37.47	-498.1	-1220	**0.987**
Sem. Dup.	0.693	-276.5	-90.21	**0.991**
No Dup.	0.650	-252.9	-7.553	**0.963**

4. Conclusion

In this work, we (a) defined clear desiderata that unlearning metric should satisfy, (b) proposed a novel data-centric LLM unlearning metric, `WaterDrum`, and (c) introduced a benchmark dataset, `WaterDrum-Ax`, which can be used with `WaterDrum` to benchmark unlearning algorithms.

References

A. Becker and T. Liebig. Evaluating machine unlearning via epistemic uncertainty. arxiv:2208.10836, 2022.

L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot. Machine unlearning. In *Proc. IEEE S&P*, pages 141–159, 2021.

Y. Cao and J. Yang. Towards making systems forget with machine unlearning. In *Proc. IEEE S&P*, pages 463–480, 2015.

CCPA. California consumer privacy act of 2018. Civil Code Title 1.81.5, 2018. URL https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375.

Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al. A survey on evaluation of large language models. *ACM TIST*, pages 1–45, 2024.

V. S. Chundawat, A. K. Tarun, M. Mandal, and M. Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proc. AAAI*, pages 7210–7217, 2023a.

V. S. Chundawat, A. K. Tarun, M. Mandal, and M. Kankanhalli. Zero-shot machine unlearning. *IEEE Trans. Information Forensics and Security*, 18:2345–2354, 2023b.

M. Duan, A. Suri, N. Mireshghallah, S. Min, W. Shi, L. Zettlemoyer, Y. Tsvetkov, Y. Choi, D. Evans, and H. Hajishirzi. Do membership inference attacks work on large language models? In *Proc. COLM*, 2024.

T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

X. Gao, X. Ma, J. Wang, Y. Sun, B. Li, S. Ji, P. Cheng, and J. Chen. VeriFi: Towards verifiable federated unlearning. *IEEE Trans. Dependable and Secure Computing*, 21(6): 5720–5736, 2024.

GDPR. Article 17 of the General Data Protection Regulation: Right to erasure ('right to be forgotten'). *Official Journal of the European Union*, 2016.

A. Golatkar, A. Achille, and S. Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proc. IEEE CVPR*, pages 9304–9312, 2020.

A. Golatkar, A. Achille, A. Ravichandran, M. Polito, and S. Soatto. Mixed-privacy forgetting in deep networks. In *Proc. IEEE CVPR*, pages 792–801, 2021.

M. M. Grynbaum and R. Mac. The Times sues OpenAI and Microsoft over A.I. use of copyrighted work. The New York Times, Dec 2023. URL https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html.

Y. Guo, Y. Zhao, S. Hou, C. Wang, and X. Jia. Verifying in the dark: Verifiable machine unlearning by using invisible backdoor triggers. *IEEE Trans. Information Forensics and Security*, 19:708–721, 2023.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. LORA: Low-rank adaptation of large language models. In *Proc. ICLR*, 2022.

Y. Hu, C. Wu, Q. Pan, Y. Jin, R. Lyu, V. Martinez, S. Huang, J. Wu, L. J. W. N. A. Clark, M. B. Raschke, Y. Zhao, and W. Zhang. Retraction note: Synthesis of γ-graphyne using dynamic covalent chemistry. *Nature Synthesis*, 3: 1311, 2024.

G. Ilharco, M. T. Ribeiro, M. Wortsman, L. Schmidt, H. Hajishirzi, and A. Farhadi. Editing models with task arithmetic. In *Proc. ICLR*, 2023.

M. Kurmanji, P. Triantafillou, J. Hayes, and E. Triantafillou. Towards unbounded machine unlearning. In *Proc. NeurIPS*, pages 1957–1987, 2024.

G. K. R. Lau, X. Niu, H. Dao, J. Chen, C.-S. Foo, and B. K. H. Low. Waterfall: Scalable framework for robust text watermarking and provenance for llms. In *Proc. EMNLP*, pages 20432—-20466, 2024.

N. Li, A. Pan, A. Gopal, S. Yue, D. Berrios, A. Gatti, J. D. Li, A.-K. Dombrowski, S. Goel, L. Phan, G. Mukobi, N. Helm-Burger, R. Lababidi, L. Justen, A. B. Liu, M. Chen, I. Barrass, O. Zhang, X. Zhu, R. Tamirisa, B. Bharathi, A. Khoja, Z. Zhao, A. Herbert-Voss, C. B. Breuer, S. Marks, O. Patel, A. Zou, M. Mazeika, Z. Wang, P. Oswal, W. Lin, A. A. Hunt, J. Tienken-Harder, K. Y. Shih, K. Talley, J. Guan, R. Kaplan, I. Steneker, D. Campbell, B. Jokubaitis, A. Levinson, J. Wang, W. Qian, K. K. Karmakar, S. Basart, S. Fitz, M. Levine, P. Kumaraguru, U. Tupakula, V. Varadharajan, R. Wang, Y. Shoshitaishvili, J. Ba, K. M. Esvelt, A. Wang, and D. Hendrycks. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *Proc. ICML*, pages 28525–28550, 2024a.

N. Li, C. Zhou, Y. Gao, H. Chen, A. Fu, Z. Zhang, and Y. Shui. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. arxiv:2403.08254, 2024b.

Y. Li, S. Bubeck, R. Eldan, A. Del Giorno, S. Gunasekar, and Y. T. Lee. Textbooks are all you need II: **phi-1.5** technical report. arxiv:2309.05463, 2023.

C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Proc. ACL Workshop on Text Summarization Branches Out*, pages 74–81, 2004.

A. Liu, L. Pan, Y. Lu, J. Li, X. Hu, X. Zhang, L. Wen, I. King, H. Xiong, and P. Yu. A survey of text watermarking in the era of large language models. *ACM Computing Surveys*, 57(2):1–36, 2024.

S. Liu, Y. Yao, J. Jia, S. Casper, N. Baracaldo, P. Hase, Y. Yao, C. Y. Liu, , X. Xu, H. Li, K. R. Varshney, M. Bansal, S. Koyejo, and Y. Liu. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, 7:181–194, 2025.

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

P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, and J. Z. Kolter. TOFU: A task of fictitious unlearning for LLMs. In *Proc. COLM*, 2024.

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

L. Y. Por, K. Wong, and K. O. Chee. UniSpaCh: A text-based data hiding method using unicode space characters. *J. Syst. Softw.*, 85(5):1075–1082, 2012.

W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer. Detecting pretraining data from large language models. In *Proc. ICLR*, 2024.

W. Shi, J. Lee, Y. Huang, S. Malladi, J. Zhao, A. Holtzman, D. Liu, L. Zettlemoyer, N. A. Smith, and C. Zhang. MUSE: Machine unlearning six-way evaluation for language models. In *Proc. ICLR*, 2025.

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *Proc. IEEE S&P*, pages 3–18, 2017.

D. M. Sommer, L. Song, S. Wagh, and P. Mittal. Athena: Probabilistic verification of machine unlearning. *PoPETs*, 2022(3):268—-290, 2022.

A. K. Tarun, V. S. Chundawat, M. Mandal, and M. Kankanhalli. Fast yet effective machine unlearning. *IEEE TNNLS*, 35(9):13046–13055, 2023.

P. Thaker, S. Hu, N. Kale, Y. Maurya, Z. S. Wu, and V. Smith. Position: LLM unlearning benchmarks are weak measures of progress. arxiv:2410.02879, 2024.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models. arxiv:2307.09288, 2023.

W. Wan, J. Wang, Y. Zhang, J. Li, H. Yu, and J. Sun. A comprehensive survey on robust image watermarking. *Neurocomputing*, 488:226–247, 2022.

Q. Wang, B. Han, P. Yang, J. Zhu, T. Liu, and M. Sugiyama. Towards effective evaluations and comparisons for LLM unlearning methods. In *Proc. ICLR*, 2025.

R. Wu, C. Yadav, R. Salakhutdinov, and K. Chaudhuri. Evaluating deep unlearning in large language models. arxiv:2410.15153, 2024.

X. Yang, J. Zhang, K. Chen, W. Zhang, Z. Ma, F. Wang, and N. Yu. Tracing text provenance via context-aware lexical substitution. In *Proc. AAAI*, pages 11613–11621, 2022.

J. Yao, E. Chien, M. Du, X. Niu, T. Wang, Z. Cheng, and X. Yue. Machine unlearning of pre-trained large language models. arxiv:2402.15159, 2024.

A. Related Works

A.1. Unlearning Metrics

Unlearning algorithms are often evaluated based on their a) unlearning effectiveness, b) utility preservation, and c) unlearning efficiency (Li et al., 2024b). We briefly discuss b) and c) as they are not the focus of this work. b) Utility preservation refers to how well the LLM maintains its performance and usability after unlearning, which can be measured with performance indicators (e.g., perplexity, accuracy) on the retain set and various downstream tasks (Chang et al., 2024). The c) efficiency of an unlearning algorithm can be assessed based on how much time and resources it saves compared to retraining from scratch (Nguyen et al., 2022; Li et al., 2024b). See Section 4 of (Liu et al., 2025) for a deeper discussion about other unlearning effectiveness, utility preservation, efficiency, and scalability metrics.

a) Unlearning effectiveness metrics. Broadly, unlearning effectiveness (or forget quality) refers to how well the LLM has removed the presence/influence of the forget set. There are a few classes of such metrics.

Utility based metrics are a form of utility-centric metrics that expect the model utility (performance indicators) when evaluated on the forget set to worsen after unlearning. LLM utility based unlearning metrics include ROUGE-L (Lin, 2004), Truth Ratio (Maini et al., 2024), and KnowMem (Shi et al., 2025). More details of their definitions can be found in App. G.2 and we have described the disadvantages of utility-centric metrics in App. F.1.

Membership inference attacks (MIA) based metrics expect the ability or probability to infer the membership of a data sample in the forget set to reduce significantly after unlearning. Some MIA-based metrics are also utility-centric, as membership inference may depend on performance indicators, such as perplexity and the log-likelihood of tokens in text data (Shi et al., 2024). However, MIA attacks (Shokri et al., 2017) have demonstrated limited success against LLMs (Duan et al., 2024), and their performance is adversely affected by the presence of similar data in the forget and retain set.

Watermarking based metrics embed signals in the forget set and expect the strength of these signals to decrease after unlearning (Li et al., 2024b). **Our algorithm `WaterDrum` falls under the category but is the first metric that works for LLMs. Existing watermarking-based unlearning metrics are designed and work only for image datasets and classification models.** For example, Guo et al. (2023) embedded invisible backdoors in images with incorrect target labels to assess the success of unlearning, measured by a drop in the success rate of backdoor attacks. Sommer et al. (2022) introduced a probabilistic verification framework for backdoors, in which users modified their data prior to submission. **We highlight the key differences of our work:** (a) These methods rely on label-based predictions and face challenges such as generalization effects, conflicting backdoor patterns, or backdoor defences. In contrast, our work focuses on adapting watermarking to LLMs, where longer and more complex output sequences provide richer signals for unlearning verification. (b) These models compromise model utility even before unlearning, especially when the forget set is large. In contrast, our framework does not significantly degrade model utility. (c) Most importantly, existing watermarking and backdoor attack-based metrics are limited to image data and cannot be directly applied as unlearning metrics for textual data due to additional challenges such as in preserving data fidelity (Guo et al., 2023; Sommer et al., 2022).

Unlearning metrics can also be classified based on whether they are **retraining-based or non-retraining-based.** Retraining is commonly viewed as the gold standard in classical unlearning settings (Cao and Yang, 2015; Golatkar et al., 2020; Bourtoule et al., 2021). This has led to various evaluation metrics that assert how closely an unlearned model approximates a retrained one, e.g., via matching performance on the forget set (Golatkar et al., 2020; Chundawat et al., 2023b) or measuring distances in weights and activations (Tarun et al., 2023; Golatkar et al., 2021; Chundawat et al., 2023a). However, retraining LLMs is often infeasible due to the scale of model parameters and the volume of pretraining data. In addition, retraining-based metrics contradict the purpose of unlearning that emphasizes the unavailability of a retrained model.

Therefore, non-retraining metrics are now more important and aligned with the growing trend of commercial LLMs that only provide black-box access. Chundawat et al. (2023a) proposes the ZRF score that captures the randomness in LLM predictions as an indicator of unlearning, while Becker and Liebig (2022) proposes to utilize model epistemic uncertainty. Yao et al. (2024) propose that a surrogate subset with the same distribution as the forget set can be employed to approximate the performance of the retrained model. However, these metrics often **overlook the LLM's ability to generalize from pre-training or the remaining retain set**. To address this, synthetic datasets, such as TOFU dataset (Maini et al., 2024),

are carefully crafted to ensure a sufficient separation between the forget and retain set. Nonetheless, **such separation and low similarity is rarely achievable in real-world scenarios**. **In this work, we address these limitations by proposing a non-retraining metric that works despite greater similarity between the forget and retain set and the generalization ability of LLMs. Additionally, our metric would work for multiple unlearning requests.** Specifically, we propose to use watermarking (Sommer et al., 2022; Guo et al., 2023; Gao et al., 2024) to handle potential similarities due to its ability to make each data point uniquely identifiable.

A.2. Comparison With Other LLM Unlearning Evaluations.

Maini et al. (2024); Shi et al. (2025) have proposed new unlearning metrics and benchmark datasets. Li et al. (2024a) proposes a multiple-choice question benchmark dataset, WMDP, to evaluate the LLM's knowledge in biosecurity, cybersecurity, and chemical security. This benchmark dataset is different from TOFU, MUSE, and ours in nature because it is specifically for knowledge editing and only contains testing data instead of training data. Wang et al. (2025) suggest that an unlearning metric should be robust against (unchanged by) red teaming scenarios (such as recovering knowledge by jail-breaking, probing, relearning) and unlearning algorithms should be compared when they achieve the same retain quality, which is realized by mixing the parameters of the LLM before and after unlearning. Wu et al. (2024) proposes a new perspective of fact unlearning and an accompanying synthetic dataset. **In contrast, we propose and satisfy a novel set of desiderata to address realistic settings, such as when the forget and retain sets have semantically similar content and when retraining is impractical. Our desiderata are not intended to be exhaustive and can complement existing benchmarks.** Lynch et al. (2024) proposes a suite of adversarial metrics to resurface forget set-related knowledge that exists in the unlearned LLMs, e.g., jailbreaking prompts, relearning (via fine-tuning and in-context learning), and latent knowledge extraction. While these metrics employ the textual similarity to the forget set in adversarial scenarios to evaluate the unlearning success, watermarking uses the signal carried in LLM outputs to detect the presence of data from the forget set.

A.3. Text Watermarking

Watermarking is an extensively studied technique for copyright protection, fingerprinting, and authentication (Wan et al., 2022; Liu et al., 2024). Watermarking consists of two main stages: embedding and detection, where the watermark must remain imperceptible and robust against removal attacks (Wan et al., 2022). Unlike digital images, where continuous signals facilitate imperceptible watermark embedding, text watermarking is more difficult due to its discrete nature and susceptibility to text modifications (Liu et al., 2024). Existing methods, such as inserting Unicode characters (Por et al., 2012) or synonym replacement (Yang et al., 2022), are often easily detectable and susceptible to word replacement. On the other hand, syntactic-based watermarking methods are often constrained by the limited choices of syntactic structures and require prior linguistic knowledge (Wan et al., 2022). Recently, LLMs have emerged as a promising watermarking tool as they can generate natural-looking text and improve watermarking robustness. Lau et al. (2024) proposed a robust text watermarking approach capable of embedding watermarks across data from multiple data owners, preserving the semantic content of the original text, and also achieving watermark robustness such that watermarks in the training data of LLMs remain detectable in the LLM output. **We build on Lau et al. (2024) framework in our work to develop our unlearning metric. Other watermarking frameworks can be considered in future works.**

B. Watermarking desiderata

In our watermarking framework, each data owner i is assigned a unique private watermark key μ_i. Our watermarking framework comprises (a) a **watermarking operator** $\mathcal{W}(d_i, \mu_i) \to d_i'$ that takes in any text data $d_i \in \mathcal{D}_i$ and produces a corresponding text data d_i' uniquely associated with watermark μ_i, and (b) a **verification operator** $\mathcal{V}(g', \mu_i)$ that takes in any text data or set of text data g' (e.g., LLM output(s)) and provides a score reflecting the likelihood of g' containing the watermark μ_i.

To satisfy our unlearning metric desiderata in Sec. 2, the watermark and verification operators used in the watermarking framework will need to satisfy a list of desiderata.

W0 Fidelity. The watermarking should have minimal impact on the semantic similarity of the original data, i.e., $d \simeq \mathcal{W}(d, \mu)$ for any watermark key μ and data $d \in \mathcal{D}_\mathcal{T}$. While this does not directly impact the unlearning metric desiderata, **W0** ensures that the watermarking process preserves the value of the data for the model owner and the metric can be deployed in practice.

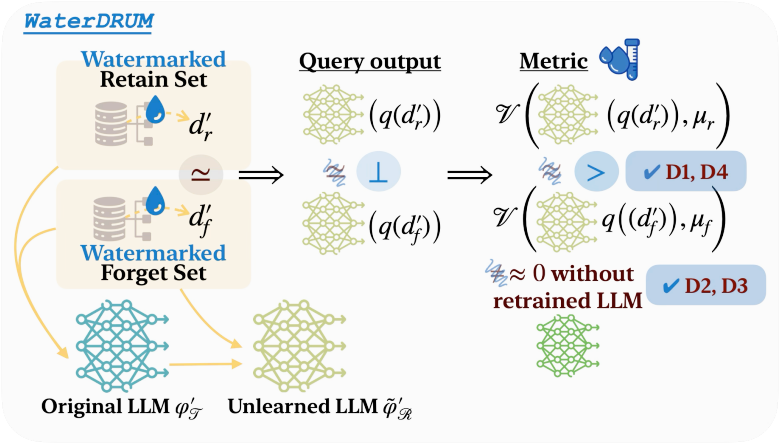


Figure 2: Unlike existing utility-centric metrics, `WaterDrum` satisfies the unlearning metric desiderata in Sec. 2. `WaterDrum` is robust to similar data as orthogonal data-specific signals are embedded in the LLM outputs that are **W1** verifiable.

W1 Verifiability. (a) The watermark should be verifiable if and only if the watermarked content is present in the LLM. In our setting, this implies that the retrained LLM should not contain the watermark of an owner f in $\mathcal{F}$ who requested to erase its data, i.e., $\mathcal{V}(\varphi_{\mathcal{R}}(q(d_f)), \mu_f) = 0$. In contrast, an LLM that has been trained on owner f's data $\mathcal{D}_f \subseteq \mathcal{D}_{\mathcal{F}}$ should have a verifiable watermark μ_f, i.e., $\mathcal{V}(\varphi_{\mathcal{F}}(q(d_f)), \mu_f) \gg 0$ for all $d_f \in \mathcal{D}_f$. (b) If every text data in $\mathcal{D}_{\mathcal{F}}$ is watermarked with the same key $\mu_{\mathcal{F}}$, the average value $\mathcal{V}(\widehat{\varphi}(q(d_f)), \mu_f)$ across all $d_f \in \mathcal{D}_{\mathcal{F}}$ for model $\widehat{\varphi}$ retrained on $\mathcal{D}_{\mathcal{R}} \bigcup \mathcal{D}_{\blacksquare}$ should be proportional to the size of the data $\mathcal{D}_{\blacksquare} \subseteq \mathcal{D}_{\mathcal{F}}$. Together, (a) and (b) support **D1** and **D2**.

W2 Overlap verifiability. The verifiability desideratum **W1** is satisfied despite the presence of other watermarks (e.g., μ_r from another owner r) in the training dataset of the model. This allows for multiple watermarks to be evaluated from the output of the same model.

We will also need additional desiderata on the watermarking process to meet the rest of the unlearning metric desiderata:

W3 Query access constraint. Data owners should be able to verify the watermark with only query access to the model. This supports **D3** with feasible and efficient unlearning evaluation.

W4 Unique key. Each data owner i's watermark key μ_i should be unique. When a forget set data owner requests full erasure of its data, the forget and retain sets will have different watermarks, with different strengths, thus supporting **D1**. Furthermore, the unique keys ensure that similar or even identical data from different owners will have different watermarks, which supports **D4**.

Fig. 2 summarizes how a framework that satisfies these desiderata can satisfy the unlearning metric desiderata in Sec. 2. Concretely, we can define a metric M' based on the verification operator.

$$M'(\varphi_{\bullet}(q(d_i)), i) \coloneqq \mathcal{V}(\varphi_{\bullet}(q(d_i)), \mu_i)\ . \tag{4}$$

To measure the influence of $\mathcal{F}$ on LLM output, we evaluate the verification score for each watermark and consider the one with the highest score: $M'(\varphi_{\bullet}(q(d_i)), \mathcal{F}) \coloneqq \max_{j \in \mathcal{F}} \mathcal{V}(\varphi_{\bullet}(q(d_i)), \mu_j)$.

C. Details on Watermarking with `Waterfall`

Watermarking and verification of the training text data was done using the `Waterfall` algorithm (Lau et al., 2024), using the default configuration of the code available on `https://github.com/aoi3142/Waterfall`. The texts were watermarked with the default LLM `meta-llama/Llama-3.1-8B-Instruct`, with watermark strength $\kappa = 2$ and perturbation key $k_p = 1$.

When watermarking for `WaterDrum-Ax`, the different data owners were assigned consecutive IDs μ, starting from 0 and incrementing by 1 for each data owner $(0, 1, 2, ...)$. For experiments involving duplicate data, we watermarked with the ID 1

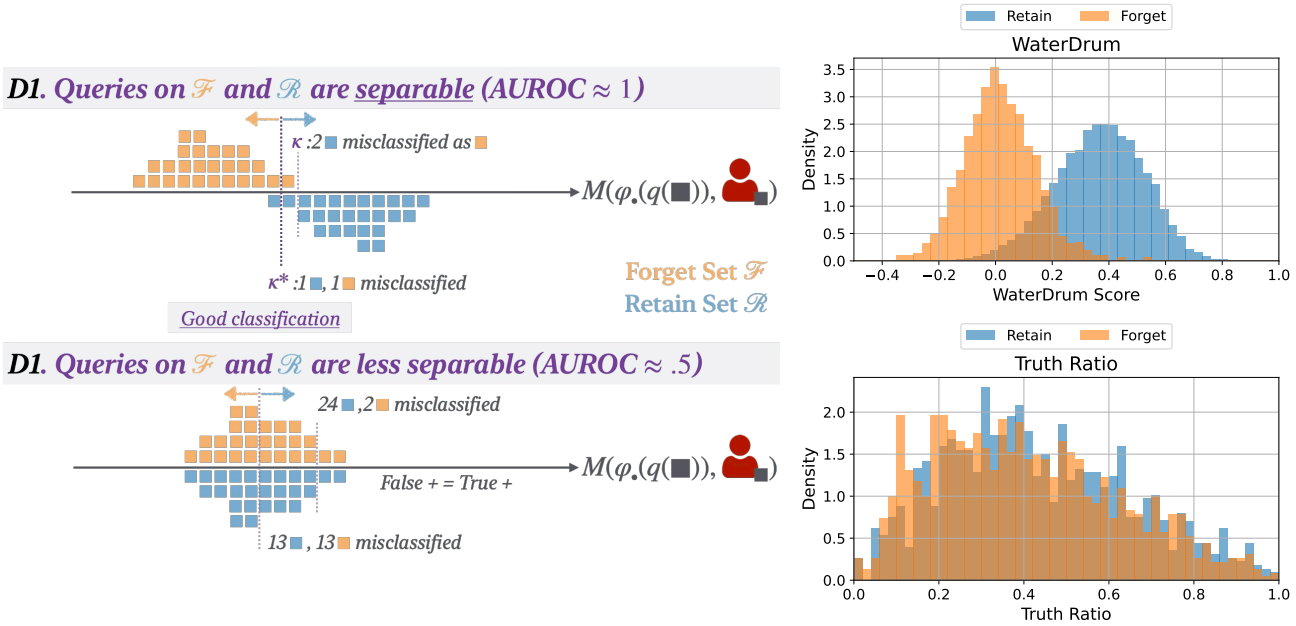


Figure 3: Each ■ represents a query. Different κ corresponds to different decision boundaries. In the top diagrams, the metric and κ^* can clearly separate queries on the forget set and the retain set. In the bottom diagrams, the queries cannot be separated clearly and for any κ, the true and false positive rates are the same. The left diagram provides a toy example to illustrate the intuition of the separability desiderata **D1**, while the right diagrams show actual plots of metrics evaluated on semantic-duplicate `WaterDrum-TOFU` from App. H.2, where `WaterDrum` exhibits clear separability over Truth Ratio.

higher than the owner index instead (i-th owner watermarked with $\mu_i = i + 1$, where i is zero-indexed). The watermark ID for the duplicate of the last owner's data is wrapped around, using $\mu_{-1} = 0$ (i.e., for majority of the duplicate experiments where there is only a single duplicate data owner duplicating the single forget class, those duplicate data were watermarked with $\mu = 0$). For the experiments with multiple data owners requesting to have their data unlearned, this simulates the situation where some owners only request for a portion of their data to be unlearned, while retaining the remaining portion of their data.

When watermarking for `WaterDrum-TOFU`, the data from the retain set was watermarked with ID $\mu = 0$ while data from the forget set was watermarked with ID $\mu = 1$. Duplicate data of the forget set were watermarked with the retain watermark, ID $\mu = 0$.

Note that as part of `Waterfall`'s watermarking process, the original texts were paraphrased with the use of an LLM. Although efforts were made to ensure that the watermarked text retains high semantic similarity with the original text (see (Lau et al., 2024) and https://github.com/aoi3142/Waterfall), we cannot guarantee the faithful reproduction of all content from the original text, nor the factual correctness of the watermarked texts. Despite this, `WaterDrum-Ax` and `WaterDrum-TOFU` still serve as suitable datasets when used for the purpose of evaluating unlearning metrics and algorithms, where the factuality of the content in the dataset is not relied upon. In practical real-world unlearning applications, additional (automated or human-involved) checks could be performed on the watermarked text to ensure accuracy and consistency to the original text (Lau et al., 2024).

D. Further Discussion on Desiderata for Unlearning Metrics

D.1. D1 Separability

A separable metric (**D1**) should be a good classifier of whether an owner's data still has influence on an unlearned LLM, in particular, the model retrained only on $\mathcal{D}_\mathcal{R}$. To illustrate the difference between a separable and non-separable metric, we provide a toy example in Fig. 3 (left). With a separable metric, an optimal threshold κ^* could be chosen where false positive and false negative classifications are minimal, as is the case for `WaterDrum` Fig. 3 (top right). However, for non-separable

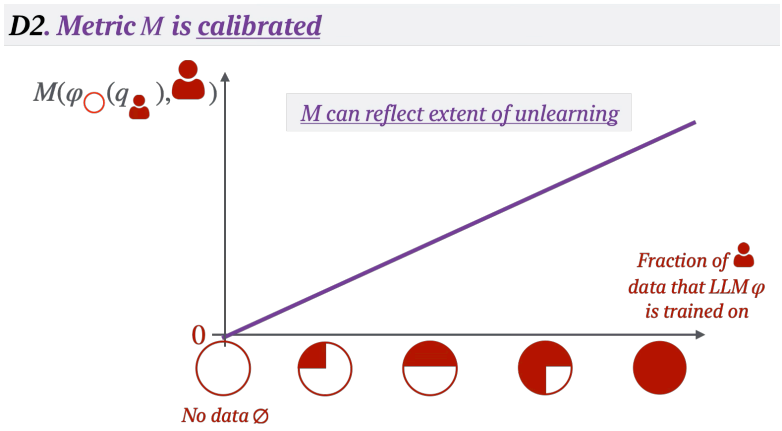


Figure 4: A calibrated metric should reflect the extent of imperfect unlearning. On the horizontal axis, we simulate using different sized fractions of the red owner's dataset. **D2** requires the metric to have 0 value when the dataset is not used and larger values when a larger fraction is used. As shown in Fig. 6 in App. H.3, `WaterDrum` is well calibration while other metrics are not.

metrics, any κ chosen would result in similar true and false positive rates, as shown in Fig. 3 (bottom right).

D.2. D2 Calibration

D2 (calibration) enables unlearning metrics to go beyond being just a binary indicator of whether an entire dataset has been unlearned, to be a meaningful continuous score of how much of a forget set $\mathcal{D}_\mathcal{F}$ has been unlearned.

- The proposed linear proportional form (Eq. (2)) of **D2** captures the desire that the unlearning metric can be directly interpreted as indicating the proportion of $\mathcal{D}_\mathcal{F}$ that has not been unlearned, given just a single calibration data point (i.e., the forget set metric evaluated on the *original* model) that is available before unlearning. This is in direct contrast to existing utility-centric metrics, which require at least one additional calibration data point (the forget set metric evaluated on the *retrained* model), violating **D3**(a) as described in App. F.1.

- Surprisingly, as seen in our experiments (Fig. 3 and Tab 3), `WaterDrum` can satisfy **D2**, enabling this intuitive and simple interpretation in the scenario of models retrained with data including varying fractions of the forget set $\frac{k}{|\mathcal{D}_\mathcal{F}|}$.

Fig. 4 provides an intuitive illustration for the calibration desideratum, where the metric score measures the extent of imperfect unlearning. We also discuss practical use cases for **D2** in App. K.2.

D.3. D4 Robustness

Similarity of data in the retain and forget set is not typically considered in other works related to unlearning, despite its prevalence in practical real world scenarios (e.g., news agencies have different news articles reporting on the same event, as illustrated in App. K.3). Our **D4** robustness desideratum directly addresses this, enforcing that the other desiderata should hold even under scenarios where similar data is present across retain and forget set.

Let $d_i \simeq d_j$ denote that text data d_i and d_j have a large *similarity score* $SS(d_i, d_j)$, e.g., computed using some semantic text similarity (STS) score, and $\mathcal{D}_i \simeq \mathcal{D}_j$ denote sets where for any $d_i \in \mathcal{D}_i$, there is a corresponding $d_j \in \mathcal{D}_j$ such that $d_i \simeq d_j$. Satisfying **D4** is challenging because the similarity of data points d_r and d_f in the respective retain and forget sets often implies that the corresponding LLM outputs will also be similar, i.e., $\varphi_\bullet(q(d_r)) \simeq \varphi_\bullet(q(d_f))$ (we empirically demonstrate this in App. I.2 for the retrained model). This makes it hard for many utility-centric metrics to satisfy both the separability and calibration desiderata and further motivate the need to adopt more data-centric unlearning metrics.

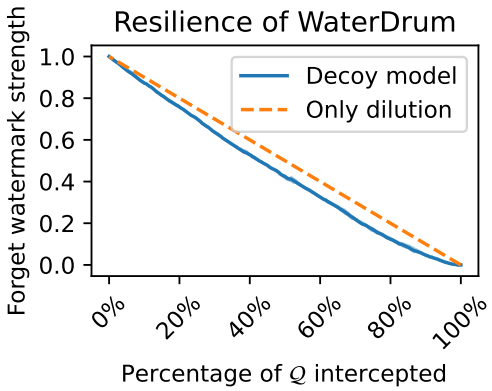


Figure 5: Plot of forget watermark strength (`WaterDrum` metric) over % of queries in Q intercepted, as the model owner increases its filtering threshold B under the threat model $\mathbb{T}$. The best possible unlearning metric against $\mathbb{T}$ will have its score decrease only proportionally (dotted orange diagonal line). `WaterDrum` achieves a similar performance, implying that the threat model requires intercepting a large proportion of queries to reduce the metric detectable by the forget set data owner. Watermark strength is scaled to 1.0 for Q before the threat model is implemented.

E. Resilience

We need to consider the realistic scenario in which the model owner's interests may not align with those of the data owners. As full unlearning is costly, the model owner is incentivized to avoid it while appearing to fulfil the data owners' erasure requests. As the model owner is aware of the metric M used, they can attempt to improve the metric through a threat model without directly performing unlearning if doing so is less costly.

To analyze this, we consider the scenario where the model owner continues to use the existing model $\varphi_{\mathcal{T}}$ instead of spending resources to unlearn $\mathcal{D}_{\mathcal{F}}$ (and produce $\widetilde{\varphi}$).

Threat model. The model owner implements the threat model $\mathbb{T}$ that involves simulating a decoy unlearned model $\breve{\varphi}$ with a gating function to intercept any query $q(d_i)$ that is received. For metrics that it could compute exactly, the model owner would filter queries that result in output with signals that indicate that the underlying model is still the full model $\varphi_{\mathcal{T}}$ with influence from the forget set $\mathcal{D}_{\mathcal{F}}$, e.g., queries $q(d_i)$ where $M(\varphi_{\mathcal{T}}(q(d_i)), f) > \kappa$ for any $f \in \mathcal{F}$, and replace $\varphi_{\mathcal{T}}(q(d_i))$ with some text $g(q(d_i), \mathcal{D}_{\mathcal{F}})$ that minimizes the metric signal. For metrics that the model owner cannot compute exactly (e.g., metrics with computation that require some information that is private to the data owner), the model owner can only resort to a proxy indicator SS that measures how similar a query output $\varphi_{\mathcal{T}}(q(d_i))$ is to the forget set $\mathcal{D}_{\mathcal{F}}$, for the filter:

$$\breve{\varphi}(q(d_i)) = \begin{cases} g(q(d_i), \mathcal{D}_{\mathcal{F}}) & \text{if } \exists d_f \in \mathcal{D}_{\mathcal{F}}, SS(\varphi_{\mathcal{T}}(q(d_i)), d_f) > B, \\ \varphi_{\mathcal{T}}(q(d_i)) & \text{otherwise} \end{cases} \tag{5}$$

with a selected threshold value B as determined by the model owner. In practice, for $g(q(d_i), \mathcal{D}_{\mathcal{F}})$, the model owner can generate an output that minimizes the score of metric M, such as by replacing it with output from another untrained model. Note that in situations where Eq. (5) is applied, the model owner will realistically only intercept queries with a large SS threshold B. Performing this for a small threshold will harm overall model performance with more decoy output replacements and will be more costly – in the extreme scenario, this approach intercepts all queries and would essentially be comparable to a full unlearning algorithm. In these cases, the metric needs to be resilient against such a threat model: i.e., exhibit **Resilience.** The metric should meet all the above desiderata, despite the model owner potentially implementing threat model $\mathbb{T}$ in Eq. (5).

Subsequently, we assess whether our `WaterDrum` metric satisfies the resilience requirement where the model owner attempts to avoid unlearning by building a decoy unlearned model $\breve{\varphi}$ (Eq. (5)). To create the impression of successful unlearning, the model owner can compute the forget set data owner $f \in \mathcal{F}$'s unlearning metric on any model output, and adjust any output with high scores to an alternative output with low scores (e.g., output from a decoy model). Such an attack would work well for all baseline metrics, since the model owner can replicate any metric computation process that is done by data owner f. Specifically, unlike `WaterDrum` where the metric score can only be computed with knowledge of the data

owner's private watermark key μ, other metrics (such as ROUGE) can be directly computed by the model owner, allowing the model owner to use the metric itself as SS. The model owner can then choose their threshold B in Eq. (5) to exactly match the data owner's threshold κ from **D1**, thereby replacing all outputs that the forget data owner would consider as influenced by the forget set. This prevents the data owner from realizing that their data still remains in the underlying model.

However, the key advantage of `WaterDrum` is that the model owner does not have the private key μ_f of data owner f to compute the metric (Eq. (3)) when building their decoy model. The model owner can only resort to some proxy indicator of similarity SS between received queries $q(d_i)$ and the forget set $\mathcal{D}_\mathcal{F}$ to decide which output it should replace to lower the `WaterDrum` metric score. The lower the threshold B it sets, the higher the chances of reducing the `WaterDrum` score, but the more output it would need to replace, increasing the cost of this attack and discouraging the model owner to avoid actual unlearning. Generating coherent replacement text without μ_f is costly, as `Waterfall` watermarks are robust to modification attacks (Lau et al., 2024) – the model owner may have to replace any intercepted output with unwatermarked text from other sources (e.g., another model) with lower quality, impacting its service to its users.

In response to the threat model, data owner f can prepare beforehand a set of queries Q that it assesses to have watermark signal above an unlearning threshold κ, i.e. $Q = \{q(d_i) | M'(\varphi_\mathcal{T}(q(d_i)), f) > \kappa\}$. In our experiment, $\mathcal{D}_\mathcal{F}$ is a set of arXiv abstracts from the `math.PR` dataset, and Q consists of other such abstracts not[2] in $\mathcal{D}_\mathcal{F}$. These queries are related to $\mathcal{D}_\mathcal{F}$ (from the same subject) in order to elicit the watermark of $\mathcal{F}$ without being actually from $\mathcal{D}_\mathcal{F}$, so as to prevent the model owner from directly recognizing and filtering them as $\mathcal{F}$'s queries. Thus, the model owner can only use the proxy indicator STS score as SS, computed between the model's generated text and all text in $d_f \in \mathcal{D}_\mathcal{F}$ to try and intercept $\mathcal{F}$'s queries.

As the model owner increases B, it potentially reduces the average watermarking score via 2 effects: (1) diluting the score by replacing the output with watermark signal by the output from unwatermarked sources, and (2) expecting a lower watermark signal from the remaining unfiltered queries that are semantically further away from the original watermarked $\mathcal{D}_\mathcal{F}$. Fig. 5 plots the `WaterDrum` metric against the percentage of intercepted queries in Q, as the threshold B is increased. Note that the unlearning metric decreases almost 1:1 with the percentage of intercepted queries, implying that the model is only relying on effect (1) with no help from effect (2), i.e., the model owner can only reduce $\mathcal{F}$'s watermark strength by indiscriminately filtering all queries that are semantically similar to $\mathcal{D}_\mathcal{F}$. This makes it very costly for the model owner to carry out the attack. For example, reducing the forget watermark strength to 0.2 requires rejecting more than 70% of the non-relevant queries Q – the model owner may favor performing actual unlearning instead.

F. The `WaterDrum-Ax` Dataset

Apart from good unlearning metrics, suitable unlearning benchmark datasets are also critical for evaluating and developing practical unlearning algorithms. However, existing benchmark datasets such as TOFU (Maini et al., 2024), MUSE (Shi et al., 2025), and WMDP (Li et al., 2024a) may not represent the realistic challenges outlined in our problem setting (Sec. 2) as they lack: (a) **Realistic forget-retain splits.** Both TOFU and MUSE only have fixed forget $\mathcal{D}_\mathcal{F}$ and retain $\mathcal{D}_\mathcal{R}$ datasets, and do not represent practical scenarios where there are multiple data owners who could decide independently whether to erase their data; and (b) **Similar data.** Both datasets do not measure and control for a range of data similarity across $\mathcal{D}_\mathcal{F}$ and $\mathcal{D}_\mathcal{R}$, and hence cannot support evaluations on unlearning metrics for **D4** and unlearning algorithms on their ability to unlearn data in $\mathcal{D}_\mathcal{F}$ that are similar to those in $\mathcal{D}_\mathcal{R}$. In fact, Thaker et al. (2024) have also identified that in these popular benchmark datasets, the forget and retain sets are disjoint (the queries on the forget set are related only to the forget set and are unrelated to the retain set) and the performance of the unlearning methods declines sharply if dependencies between both sets are introduced. This underscores the importance of considering datasets with less disjoint and more similar data.

To address these limitations, we introduce a complementary unlearning benchmark dataset, `WaterDrum-Ax`. `WaterDrum-Ax`, comprising arXiv paper abstracts across various categories published after the release of the Llama-2 model, includes (a) abstracts from the 20 most popular academic subject categories to represent 20 different data owners that can be freely assigned to define $\mathcal{D}_\mathcal{F}$ and $\mathcal{D}_\mathcal{R}$; and (b) varying levels of data similarity ranging from exact duplicates to paraphrased versions of the abstracts that can be used across $\mathcal{D}_\mathcal{F}$ and $\mathcal{D}_\mathcal{R}$ to support evaluation of the *practicality* and *resilience* of the unlearning metrics, especially the assessment of **D4** on robustness to similar data. Overall, `WaterDrum-Ax` contains 400 abstracts for each category, aggregating to a total of 8000 data points in `WaterDrum-Ax`. These abstracts have an average length of 260 tokens, which is considerably longer than that of TOFU (Maini et al., 2024) (59 tokens). For

[2]For simplicity, in our experiments the data owner does not include queries based on $\mathcal{D}_\mathcal{F}$ in Q as it can assume that the model owner would definitely filter any output $\varphi_\mathcal{T}(q_\mathcal{F})$ based on it.

licensing information on individual papers in the dataset, see https://arxiv.org/help/license.

The `WaterDrum-Ax` benchmark dataset can be used to: (i) evaluate unlearning metrics based on the desiderata introduced in Sec. 2, and (ii) evaluate unlearning algorithms on effective and practical metrics identified in (i). The empirical evaluations in App. H focus on (i) but include some preliminary results on (ii) in App. H.4. We leave more systematic investigations of (ii) to future work.

F.1. Challenges for utility-centric unlearning metrics

Utility-centric unlearning metrics have evaluated the effectiveness of unlearning based on model utility (performance) indicators, such as verbatim memorization, perplexity, and accuracy on downstream tasks. Performance indicators P have compared the unlearned LLM $\widetilde{\varphi}$'s output to queries (e.g., $\widetilde{\varphi}(q_{\mathcal{F}})$ on the forget set) to the original data (e.g., $\mathcal{D}_{\mathcal{F}}$). We describe several types of utility-centric unlearning metrics in App. G.2.

However, such performance indicators P do not meet our required desiderata for the metric M (Sec. 2). First, **D3**(a) does not allow retraining the LLM. Without retraining, the reference value $P(\varphi_{\mathcal{R}}(q_{\mathcal{F}}), \mathcal{D}_{\mathcal{F}})$ of the perfectly unlearned LLM (i.e., retrained LLM $\varphi_{\mathcal{R}}$) cannot be determined and thus cannot be used to offset the metric to produce a value close to 0 when the forget set $\mathcal{D}_{\mathcal{F}}$ is perfectly unlearned (e.g., it is not possible to define and compute M as $P(\widetilde{\varphi}(q_{\mathcal{F}}), \mathcal{D}_{\mathcal{F}}) - P(\varphi_{\mathcal{R}}(q_{\mathcal{F}}), \mathcal{D}_{\mathcal{F}})$). Thus, without retraining, P does not satisfy **D2**, making it difficult to identify successful unlearning of the forget set. Next, when there are similar data present in the forget and retain sets (**D4**), we observe that any unlearned LLM $\widetilde{\varphi}$ (e.g., the retrained LLM $\varphi_{\mathcal{R}}$) tends to produce similar outputs to queries on both sets, that is, $\widetilde{\varphi}(q_{\mathcal{F}}) \simeq \widetilde{\varphi}(q_{\mathcal{R}})$, as empirically verified in App. I.2. As the performance indicators largely depend on direct comparisons with the LLM outputs, their corresponding values will also be similar, i.e., $P(\widetilde{\varphi}(q_{\mathcal{F}}), \mathcal{D}_{\mathcal{F}}) \approx P(\widetilde{\varphi}(q_{\mathcal{R}}), \mathcal{D}_{\mathcal{R}})$. We will show in App. H that this leads to utility-centric metrics failing to satisfy **D1** when the data from the forget and retain sets are highly or moderately similar. The failure arises because expecting poor predictions on the forget set and a low $P(\widetilde{\varphi}(q_{\mathcal{F}}), \mathcal{D}_{\mathcal{F}})$ overlooks the generalization capability of LLMs (Liu et al., 2024). Table 1 presents a comparison of our `WaterDrum` and existing metrics based on the proposed desiderata in Sec. 2.

G. Experimental setup.

We conduct our experiments on NVIDIA L40 and H100 GPUs. Evaluation is averaged across 3 random seeds $\{41, 42, 43\}$. Text generation from the different models used temperature $= 1$, top-p $= 1$, top-k left as the LLM vocabulary size.

G.1. Datasets and Training Hyperparameters.

Our primary experiments were conducted on the `WaterDrum-Ax` (App. F) and `WaterDrum-TOFU` (derived from TOFU (Maini et al., 2024) (MIT License), details in App. C) benchmark datasets, with the pre-trained Llama-2-7B (Touvron et al., 2023) as the base model:

`WaterDrum-Ax`. We finetune the bfloat16-pretrained Llama-2-7B model from Hugging Face[3] using LoRA ($r = 8$, $\alpha = 32$) with batch size 128 , 20 training epochs, learning rate 1e$-$3. Additionally, we finetune the bfloat16-pretrained Phi-1.5 model (detailed in App. J.2) with the same settings. We have considered these two models as they are representative of the recent LLMs, different in terms of model architectural details, and span different model scales.

`WaterDrum-TOFU`. We finetune the bfloat16-pretrained Llama-2-7B-chat model from Hugging Face[4] using LoRA ($r = 8$, $\alpha = 32$) with batch size 128 , 10 training epochs, learning rate 1e$-$4.

The models were finetuned with different data subsets under various settings. For unlearning, we consider the last 1 class from `WaterDrum-Ax` and the last 10% data from `WaterDrum-TOFU` as the forget sets, and use a batch size of 32. While we conduct our experiments using LoRA as in other LLM unlearning works (Maini et al., 2024; Shi et al., 2025), we also demonstrated that `WaterDrum` applies to full parameter fine-tuning in App. J.1. We also conducted experiments on other LLMs (Li et al., 2023) detailed in App. J.2.

[3] `https://huggingface.co/meta-llama/Llama-2-7b-hf`.

[4] `https://huggingface.co/meta-llama/Llama-2-7b-chat-hf`.

G.2. Baseline Metrics.

For baselines, we compare `WaterDrum` against recent and commonly adopted unlearning metrics: ROUGE-L (Lin, 2004), Truth Ratio (Maini et al., 2024), KnowMem (Shi et al., 2025) and MIA (Shi et al., 2024). Other than our metric `WaterDrum`, we consider these other baseline metrics as utility-centric unlearning metrics.

- **ROUGE-L**: measures the longest common subsequence between the generated text and a reference text. This serves as a surrogate for the generation quality for the `WaterDrum-Ax` dataset and the question-answering accuracy for the `WaterDrum-TOFU` dataset. For the `WaterDrum-Ax` dataset, we prompted the LLM with the first 50 tokens of the training dataset for the LLM to perform completion generation. For the `WaterDrum-TOFU` dataset, we prompted the LLM with the questions, using the LLM's prompt format. To calculate the metric score, we follow Shi et al. (2025); Maini et al. (2024) in computing the ROUGE-L recall scores (Lin, 2004) to compare the LLM response with the training data as ground truth. We generated 10 outputs for each prompt, and the mean score for the 10 generations was taken.
- **Truth Ratio**: measures the probability of generating a correct answer versus a wrong answer as an indicator of whether the LLM still memorizes the knowledge to be unlearned on the `WaterDrum-TOFU` dataset. Following Maini et al. (2024), for each given question, we compute the ratio by dividing the averaged probabilities of multiple wrong answers by the probability of a paraphrased true answer.
- **KnowMem**: measures the ROUGE score of QA pairs related to the training data to measure the LLM memorization of the knowledge on the `WaterDrum-Ax` dataset. Following (Shi et al., 2025), we use GPT-4 to create a question-answering evaluation set with 8000 QA pairs based on the abstracts in the `WaterDrum-Ax` dataset and measure the ROUGE score between the LLM's generated response to the questions and the ground truth answers.
- **MIA**: measures the difference in predictive distribution between two models to measure privacy leakage from unlearning. Specifically, we employ the state-of-the-art Min-40% attack (Shi et al., 2024) based on the loss on the forget set and holdout set, and compute AUROC of discriminating the losses.
- **`WaterDrum`**: We also use our proposed watermark metric and compare the results against the above-mentioned baseline evaluation metrics. We used the same generation setup as that in ROUGE-L for `WaterDrum`, and evaluated the watermark strength of only the generated output excluding the prompt.

For ease of comparability, all metrics are scaled such that their score when evaluated on the original model $\varphi_{\mathcal{T}}$ (which is accessible to the data owners before unlearning) is 1.0. As our `WaterDrum` framework involves watermarking the training data $\mathcal{D}_{\mathcal{T}}$ (**P1**), the models finetuned on this watermarked dataset differ slightly from the dataset used for other metrics. However, their performance remains comparable, as `Waterfall` satisfies desideratum **W0** (as shown in App. I.1).

G.3. Details on data duplication

We examine 3 representative scenarios where there exists extra data $\mathcal{D}_s$ that is similar to $\mathcal{D}_{\mathcal{F}}$ with different SS: **(a) Exact duplication:** $\mathcal{D}_s$ is an exact copies of $\mathcal{D}_{\mathcal{F}}$, hence we make $\mathcal{D}_s$ as a copy of $\mathcal{D}_{\mathcal{F}}$. This marks the highest similarity with STS $= 1.00$ and ROUGE $= 1.00$. **(b) Semantic duplication:** $\mathcal{D}_s$ is a paraphrased version of $\mathcal{D}_{\mathcal{F}}$ with the same semantic meaning. We use GPT-4 to paraphrase $\mathcal{D}_{\mathcal{F}}$ and obtain $\mathcal{D}_s$. In this case, $\mathcal{D}_s$ has STS $= 0.97$, ROUGE $= 0.69$ on `WaterDrum-Ax`, and STS $= 0.96$, ROUGE $= 0.60$ on `WaterDrum-TOFU`. We also consider the standard scenario when there is **(c) No duplication** at all in the dataset, i.e., $\mathcal{D}_s = \emptyset$.

We then finetune 3 models on the `WaterDrum-Ax` dataset $\mathcal{D}^s_{\mathcal{R}} = \mathcal{D}_s \bigcup \mathcal{D}_{\mathcal{R}}$ during finetuning, corresponding to the 3 different levels of similarity. Note that since $\mathcal{D}_s$ is from a different data owner than $\mathcal{D}_{\mathcal{F}}$, we embed different watermarks for $\mathcal{D}_s$ and $\mathcal{D}_{\mathcal{F}}$ for the evaluation of our `WaterDrum` (details in App. C). Subsequently, we adopt the set of considered unlearning methods (including retraining the model on just the retain set $\mathcal{D}^s_{\mathcal{R}}$) to remove $\mathcal{D}_{\mathcal{F}}$ while retaining $\mathcal{D}^s_{\mathcal{R}}$.

G.4. Details on calibration

In our experiments, we simulated varying sizes of subsets of the forget set by partitioning the forget set sequentially into 10 partitions, and retraining LLMs with by incrementally including partitions (and the retain set) in the training set of the retrained LLMs, i.e., using the first $0\%, 10\%, 20\%, \ldots, 100\%$ of $\mathcal{D}_{\mathcal{F}}$ as $\mathcal{D}_{\text{◓}}$ when retraining the LLMs on $\mathcal{D}_{\mathcal{R}} \bigcup \mathcal{D}_{\text{◓}}$. We

observed in App. H.3 that WaterDrum satisfies the calibration desiderata under this method of partitioning, and believe that in general, this would hold in expectation for randomly sampled fixed-size subsets of the forget set.

G.5. Baseline Unlearning Algorithms

In our experiments, we have adopted several popular baseline unlearning algorithms detailed as follows:

- **Retrain**: Directly retraining the LLM from the base LLM on the retain set. The retrained model usually serves as the golden standard for other unlearning methods.
- **Finetune**: Continually training the LLM on the retain set for 1 or several epochs. This method assumes that the LLM naturally forgets about the forget set as learning progresses on the retain set. In this paper, we finetune for 1 epoch using a learning rate of .001.
- **KL Minimization (KL)** (Maini et al., 2024): Concurrently maximizing the prediction loss on the forget set and minimizing the Kullback-Leibler divergence of predictions on the retain set to the original model. We ran KL minimization for 5 unlearning epochs.
- **SCRUB** (Kurmanji et al., 2024): Maximizing the Kullback-Leibler divergence of predictions on the forget set to the original model, while minimizing the prediction loss and divergence on the retain set. The optimization process alternates between maximization steps and minimization steps. In our experiments, we ran 3 maximization and minimization epochs.
- **Direct Preference Optimization (DPO)** (Maini et al., 2024): For question-answering tasks, encouraging responses such as "I don't know" on the forget set, while simultaneously minimizing the prediction loss on the retain set. Note that this method is not compatible with completion tasks, and is omitted for the WaterDrum-Ax dataset. We ran 5 unlearning epochs for DPO.
- **Task Vector (TV)** (Ilharco et al., 2023): Subtracting the parameters of the model retrained only on the forget set from the original model. In the experiments, we finetune the model on the forget set for 5 epochs.

Note that we excluded Gradient Ascent (Maini et al., 2024) from the unlearning algorithms considered, as they have been shown to perform poorly in other works, where the LLM output becomes gibberish or random words (Maini et al., 2024).

H. Experiments

H.1. Practicality desiderata (D3, D4)

We first evaluate WaterDrum and the baseline metrics on the effectiveness and practicality desiderata, **D1**-**D4**, as we have outlined in Sec. 2. To do so, we establish experimental settings that mimic the real-life scenarios described in the practicality desiderata **D3** and **D4**. Then, under these settings, we evaluate the effectiveness of various metrics based on **D1** and **D2**, by considering how they evaluate the perfect unlearning algorithm – retraining the model on only the retain set to generate $\varphi_{\mathcal{R}}$, which is guaranteed to contain no influence from the forget set $\mathcal{D}_{\mathcal{F}}$ by construction.

Feasibility (D3). All of the baseline metrics (ROUGE-L, Truth Ratio, KnowMem and MIA) typically require retraining a model $\varphi_{\mathcal{R}}$ with just the retain set $\mathcal{D}_{\mathcal{R}}$ to get reference values $M(\varphi_{\mathcal{R}}(q_{\mathcal{F}}), \mathcal{F})$, and hence violate **D3**(a). In our experiments, we show how the effectiveness of these metrics gets significantly impacted without access to $\varphi_{\mathcal{R}}$. In contrast, WaterDrum does not require $\varphi_{\mathcal{R}}$ as it naturally has $M'(\varphi_{\mathcal{R}}(q'_{\mathcal{F}}); \mathcal{F}) = 0$ since it satisfies **W1**. In addition, the computation of the MIA metric requires logit-access, which violates **D3**(b). However, for evaluation purposes, we grant MIA logit-access in our experiments.

Robustness to similar data (D4). We establish the settings to assess the robustness of the unlearning metrics to similar data by injecting a small amount of data $\mathcal{D}_s \simeq \mathcal{D}_{\mathcal{F}}$ into $\mathcal{D}_{\mathcal{R}}$, i.e., the retain set is augmented ($\mathcal{D}^s_{\mathcal{R}} = \mathcal{D}_s \bigcup \mathcal{D}_{\mathcal{R}}$) with some data points that are similar to $\mathcal{D}_{\mathcal{F}}$. We consider two such scenarios: (a) **Exact duplication.** Data points in $\mathcal{D}_s$ are exact copies of those in $\mathcal{D}_{\mathcal{F}}$, ($\mathcal{D}_s = \mathcal{D}_{\mathcal{F}}$) and (b) **Semantic duplication.** Data points in $\mathcal{D}_s$ are paraphrased version of $\mathcal{D}_{\mathcal{F}}$, ($\mathcal{D}_s \simeq \mathcal{D}_{\mathcal{F}}$). In addition, we consider the case where there is (c) **no duplication** of $\mathcal{D}_{\mathcal{F}}$ data points in $\mathcal{D}_{\mathcal{R}}$, ($\mathcal{D}_s = \emptyset$). Additional implementation details are in App. G.3.

Table 4: AUROC (± across 3 seeds) of metrics for different levels of similarity for the `WaterDrum-TOFU` dataset and `WaterDrum-Ax` dataset. `WaterDrum`'s AUROC remains near 1.0 even when similar data exists.

	WaterDrum-TOFU			WaterDrum-Ax		
Similarity	ROUGE	Truth Ratio	WaterDrum	ROUGE	KnowMem	WaterDrum
Exact Duplicate	0.510±0.007	0.508±0.008	**0.926±0.027**	0.334±0.005	0.492±0.005	**0.957±0.008**
Semantic Duplicate	0.798±0.001	0.472±0.054	**0.954±0.001**	**0.960±0.002**	0.450±0.007	**0.963±0.001**
No Duplicate	**0.908±0.005**	0.747±0.011	**0.928±0.026**	**0.974±0.001**	0.491±0.008	**0.965±0.002**

H.2. Separability desideratum (D1)

To assess whether the unlearning metrics satisfy **D1**, note that the left-hand side expression $\mathbb{P}[M(\varphi_{\mathcal{R}}(q(d_r)), r) > M(\varphi_{\mathcal{R}}(q(d_f)), f)]$ in Eq. (1) corresponds to the definition of the AUROC of the metric M in distinguishing between $\mathcal{R}$ and $\mathcal{F}$ (Fawcett, 2006). Hence, we can compute the AUROC of various unlearning metrics with the retrained model $\varphi_{\mathcal{R}}$, and assess if the AUROC ≈ 1. We exclude MIA from this experiment because it focuses solely on assessing privacy leakage based on distributional differences between forget and holdout sets, without considering the retain set.

Table 4 shows the AUROC of the metrics on the `WaterDrum-TOFU` dataset under various duplicate settings. Notably, `WaterDrum` is the only metric that consistently achieves AUROC > 0.9 and close to 1, hence satisfying **D1**. In contrast, the other metrics' performance degrades significantly in the duplicate settings, with AUROC dropping to around 0.5, which means the metrics are no better than random assignment in separating the forget and retain sets. Furthermore, note that Truth Ratio only achieves an AUROC of about 0.75 even in the 'no duplicate' setting, indicating that it does not satisfy **D1** under the basic settings.

Empirical results on `WaterDrum-Ax` (Table 4) show similar trends, with `WaterDrum` consistently performing well and KnowMem encountering difficulties in all settings. ROUGE performs poorly under the 'exact duplicate' setting where only 5% of the augmented retain set are duplicates of the forget set. While it performs well for the 'semantic duplicate' settings in this experiment, this occurs because the mean ROUGE score between $\mathcal{D}_s$ and $\mathcal{D}_{\mathcal{F}}$ is still low (≈ 0.65) although the mean semantic similarity score of $\mathcal{D}_s$ and $\mathcal{D}_{\mathcal{F}}$ is high (STS $= 0.94$). The lower ROUGE score implies that the text has already been heavily paraphrased such that the 'semantic duplicate' setting is effectively closer to the 'no duplicate' setting for ROUGE in this experiment. Milder forms of perturbation for this dataset would likely make its degradation of performance on **D1** more apparent.

H.3. Calibration desideratum (D2)

Next, we assess whether the unlearning metrics meet the calibration desideratum, as defined in Eq. (2). Failing to meet this desideratum implies that the metrics cannot indicate the extent to which the forget set has been unlearned in a given model. We evaluate this by first producing LLMs retrained on $\mathcal{D}_{\mathcal{R}} \bigcup \mathcal{D}_{\text{◓}}$ with varying size k of the subset $\mathcal{D}_{\text{◓}}$ included. We then compute the unlearning metrics for each retrained model and plot calibration curves showing how the metrics vary with different k. To quantify how well the metrics satisfy Eq. (2), we can compute the R^2 value for the best-fit line with the vertical intercept at 0, since a calibrated metric should be proportional to $k/|\mathcal{D}_{\mathcal{F}}|$ and have $M(\varphi_{\mathcal{R}}(q_{\mathcal{F}}), \mathcal{F}) = 0$ when no data from the forget set is used for training. R^2 values close to 1 imply that the metrics are well calibrated, while large negative values occur when the metrics produce similar, instead of proportional, values for varying percentages.

Fig. 6 shows the calibration curves for the various unlearning metrics, and Table 5 the corresponding R^2 values, under the various duplicate settings for the `WaterDrum-Ax`. Note that `WaterDrum` is the only metric that is calibrated across all settings, and can represent the percentage of forgotten data remaining in the unlearned model. In fact, the rest of the unlearning metrics perform poorly across *all settings*, including the basic 'no duplicate' setting — they cannot be used to tell when $\mathcal{D}_{\mathcal{F}}$ is perfectly unlearned as $M(\varphi_{\mathcal{R}}(q_{\mathcal{F}}), \mathcal{F}) \neq 0$.

The results demonstrate the strong reliance of the baseline methods on access to the retrained model. These methods fail to quantify the extent of unlearning, or even indicate the success of unlearning, without knowledge of the reference value that a perfectly unlearned model (e.g., $\varphi_{\mathcal{R}}$) is expected to have. This reliance is impractical as unlearning algorithms were designed precisely to approximate retrained models that are infeasible to obtain. Fig. 12 and Table 10 in App. L.2.1 show similar results for the `WaterDrum-TOFU` dataset, where all baseline metrics fail to meet the calibration desideratum for all settings, including the 'no duplicate' setting. More results are provided in App. L.1.1.

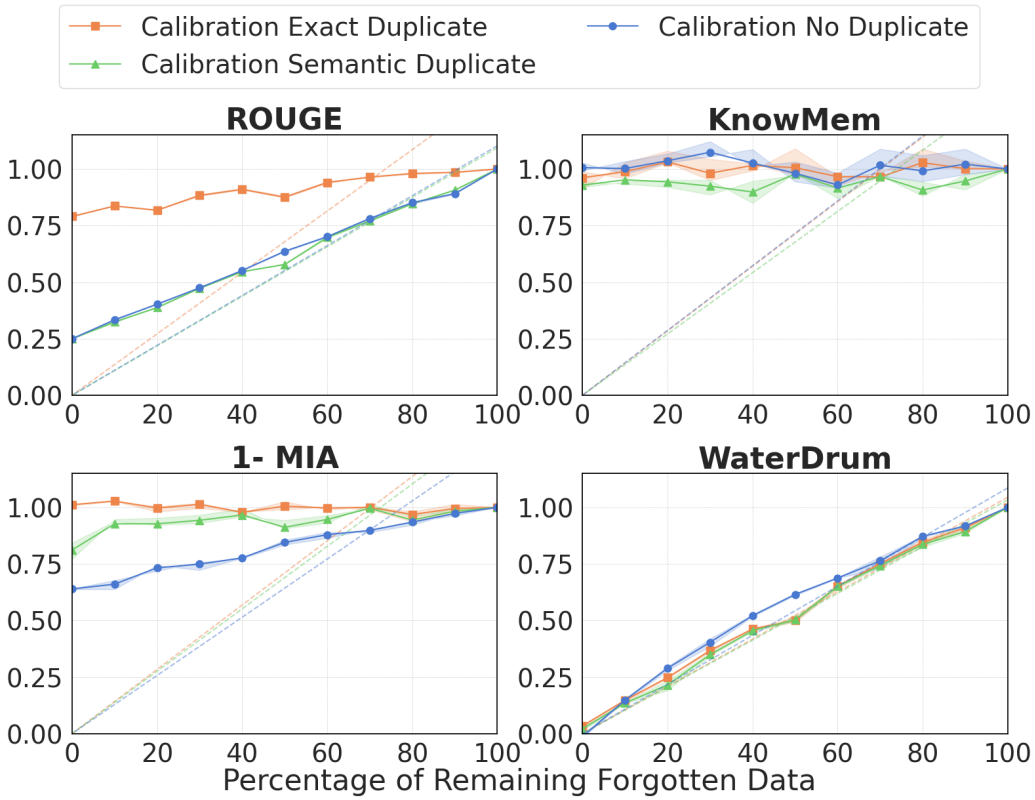


Figure 6: Plots of unlearning metrics against the % of $\mathcal{D}_\mathcal{F}$ remaining in the retrained model, under settings with different levels of data similarity for the `WaterDrum-Ax` dataset. Note that apart from `WaterDrum`, none of the other metrics are calibrated and satisfy **D2**. With `WaterDrum`, the results are very close for the three settings with different levels of similarity. Associated R^2 are in Table 5.

H.4. Benchmarking unlearning algorithms

Finally, we provide a basic illustration of how we could use `WaterDrum` to benchmark unlearning algorithms. A `WaterDrum` evaluation plot shows the unlearning algorithms evaluated on two axes: $M'(\widetilde{\varphi}(q_\mathcal{R}), \mathcal{R})$ on the x-axis and $M'(\widetilde{\varphi}(q_\mathcal{F}), \mathcal{F})$ on the y-axis that measure the retain and forget watermark strength, respectively, on an unlearned model $\widetilde{\varphi}$. The original model $\varphi_\mathcal{T}$, which contains both $\mathcal{D}_\mathcal{F}$ and $\mathcal{D}_\mathcal{R}$, is at the top right corner, while the retrained model $\varphi_\mathcal{R}$, which only contains $\mathcal{D}_\mathcal{R}$, is at the bottom right corner. It is expected for the metric evaluated for $\mathcal{D}_\mathcal{R}$ on the retrained LLM $\varphi_\mathcal{R}$ to be approximately the same as that of the original LLM $\varphi_\mathcal{T}$, as the retain set is not removed in both the original and the retrained model. In this plot, the closer the algorithms are to the retrained model, the better they are at both unlearning $\mathcal{D}_\mathcal{F}$ while retaining the influence of $\mathcal{D}_\mathcal{R}$.

Fig. 7 shows the `WaterDrum` evaluation plot for several unlearning algorithms (Finetune, KL Minimization (KL) (Maini et al., 2024), Task Vector (TV) (Ilharco et al., 2023), SCRUB (Kurmanji et al., 2024); details are in App. G.5). Note that most algorithms are still far from reaching the retrained model performance. The KL and TV algorithms achieve good unlearning quality but significantly compromise the retain set's influence and model's overall utility, while Finetune and Scrub maintain some retain performance but do not achieve the best unlearning quality. We also performed preliminary experiments for the cases with multiple parties and duplicate data in App. L.3.

I. Additional experiments

I.1. Quantitative evidence that watermarking with `Waterfall` does not degrade LLM performance

Our `WaterDrum` framework lays out desiderata for compatible watermarking methods (App. B), including fidelity (**W0**). We chose to use `Waterfall` (Lau et al., 2024) as their paper already presented extensive empirical results showing that its

Table 5: R^2 of the best fit line (dotted in Fig. 6) for metrics under different levels of similarity on the `WaterDrum-Ax` dataset. `WaterDrum` is very well linearly calibrated across the settings, with the highest R^2 value.

Similarity	ROUGE	KnowMem	MIA	`WaterDrum`
Exact Duplicate	-37.47	-498.1	-1220	**0.987**
Semantic Duplicate	0.693	-276.5	-90.21	**0.991**
No Duplicate	0.650	-252.9	-7.553	**0.963**

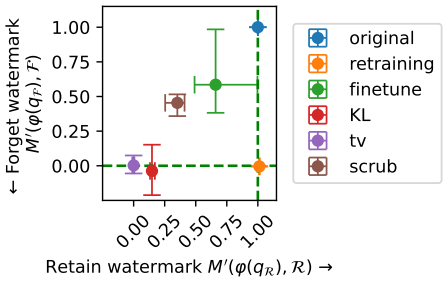


Figure 7: Using `WaterDrum`, we benchmark unlearning methods on `WaterDrum-Ax`; green lines denote optimal unlearning values.

Table 6: Semantic similarity of q_f and q_s from the `WaterDrum-Ax` dataset. For reference, the STS score of texts from the same category is 0.67.

Similarity of query	STS score of query output
Exact Duplicate	0.96
Semantic Duplicate	0.87

watermarking process has minimal degradation on LLM performance (App H.3).

Nonetheless, we have confirmed `Waterfall`'s fidelity for our experiments by comparing the trained LLM's performance when trained on the un/watermarked data using truth ratio (Maini et al., 2024), which computes each LLM's probability of generating the correct answer compared to a set of wrong answers perturbed from the correct answer.

Our results show that on the `WaterDrum-TOFU` dataset, the truth ratio of un/watermarked LLMs are very similar, at 0.5143 and 0.5163, respectively. This shows that watermarking has minimal impact on the LLM's performance.

I.2. Similarity of output in retrained LLM

Following the setup in App. H.1, under the setting where the retain set ($\mathcal{D}^s_\mathcal{R} = \mathcal{D}_s \bigcup \mathcal{D}_\mathcal{R}$) contains some data points that are similar to the forget set ($\mathcal{D}_s \simeq \mathcal{D}_\mathcal{F}$), we verify that output of the LLM $\widetilde{\varphi}^s$ trained on the retained set $\mathcal{D}^s_\mathcal{R}$ are similar for the duplicate queries $\widetilde{\varphi}^s(q_\mathcal{F}) \simeq \widetilde{\varphi}^s(q_s)$.

We empirically verify the similarity by evaluating the STS scores between the outputs of the forget query $q_\mathcal{F}$ and the retain query q_s. As shown in Table 6, the mean STS scores are 0.96 and 0.87 for exact and semantic duplicates, respectively. For comparison, the STS score of query outputs from the same `WaterDrum-Ax` category (i.e., outputs for queries from the same arXiv category such as the `math.PR` subject) only have a mean STS score of 0.67. This shows that the query outputs from the duplicate queries are very similar, much more so than queries from the same subject.

I.3. Similar metrics score across data

We verify that data points from $\mathcal{D}_s$ and $\mathcal{D}_f$ with similar semantics will have similar metric scores ($M(\varphi_\mathcal{R}(q_s), \mathcal{D}_s) \simeq M(\varphi_\mathcal{R}(q_\mathcal{F}), \mathcal{F})$). We use our `WaterDrum` to measure the metric scores on data points from $\mathcal{D}_s$ and $\mathcal{D}_f$ for the `WaterDrum-Ax` dataset when unlearning 1 class. Fig. 8 shows a histogram plot of the metric scores for the two different subsets with similar semantics. This verifies that the distributions of metric scores from the two subsets are similar.

J. Ablations

J.1. Evaluation on full parameter fine-tuning

The majority of the experiments were conducted using LoRA (Hu et al., 2022), following the setting in other LLM unlearning works (Maini et al., 2024; Shi et al., 2025). To show that `WaterDrum` is also applicable when used for full parameter fine-tuning, we conducted experiments for the separability (**D1**) and calibration (**D2**) desiderata with varying levels of similarity for the `WaterDrum-Ax` dataset.

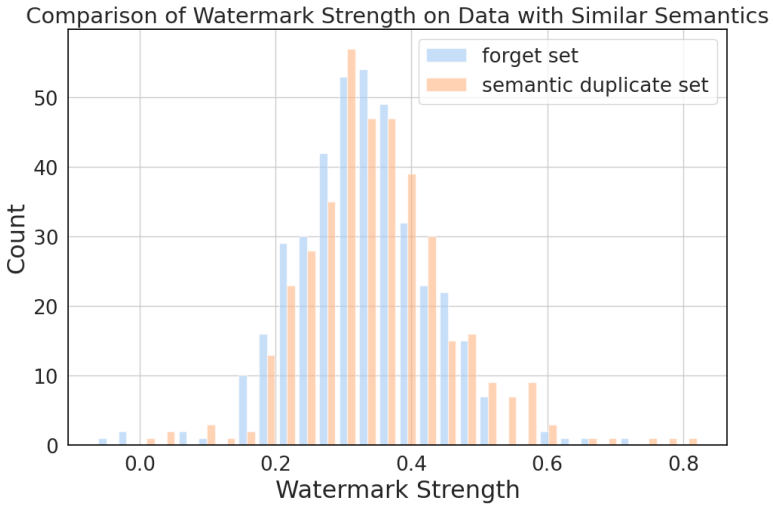


Figure 8: Count of data with different watermark strengths measured on $\mathcal{D}_f$ and $\mathcal{D}_s$ (with similar semantics) for the `WaterDrum-Ax` dataset when unlearning 1 class. The result shows that metric scores from the two sets have a similar distribution.

Table 7: AUROC of metrics for different levels of similarity for the `WaterDrum-Ax` dataset (right). `WaterDrum` 's AUROC remains near 1.0 even when similar data exists.

Similarity		ROUGE	KnowMem	`WaterDrum`
Exact Duplicate	Full	0.335	0.497	0.990
	LoRA	0.334	0.492	0.957
Semantic Duplicate	Full	0.965	0.447	0.990
	LoRA	0.960	0.450	0.963
No Duplicate	Full	0.984	0.481	0.991
	LoRA	0.974	0.491	0.965

For full parameter fine-tuning, we used a learning rate of 1e-4 and trained for 10 epochs. Note that due to the high computational cost of full parameter fine-tuning, we only report the results for one seed, while the results for LoRA are averaged across three different seeds.

Table 7 and Table 8 show that `WaterDrum` performs better than other metrics, for both LoRA and full parameter fine-tuning. The LoRA and full-parameter fine-tune results are very similar for `WaterDrum` across the experiments, showing that `WaterDrum` consistently achieves the best performance across different settings.

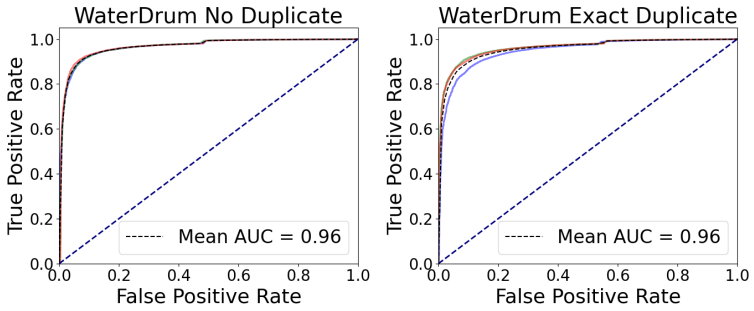
J.2. Evaluation on other models

We have also evaluated our `WaterDrum` on Phi-1.5[5] to verify its adaptability to different LLMs. Figs. 9a and 9b illustrate the AUROC and calibration for the settings of 'no duplicate' and 'exact duplicate'. The result on Phi-1.5 aligns with our main experiments using Llama2-7B and meets the proposed desiderata. This validates our `WaterDrum`'s adaptability to different LLMs, which guarantees its real application potential.

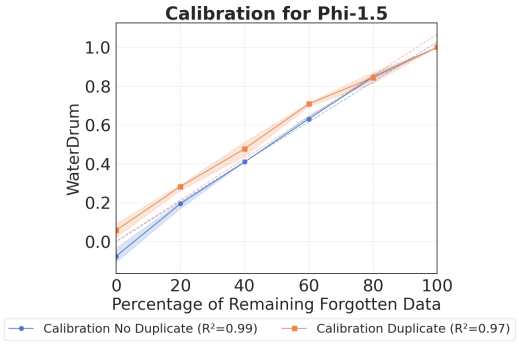
[5] `https://huggingface.co/microsoft/phi-1_5`.

Table 8: R^2 of the best fit line for various metrics under different levels of similarity for the `WaterDrum-Ax` dataset. `WaterDrum` is very well linearly calibrated across the settings, with the highest R^2 value.

Similarity		ROUGE	KnowMem	MIA	`WaterDrum`
Exact Duplicate	Full	-5059	-981.5	-4.774	0.984
	LoRA	-37.47	-498.1	-1220	0.987
Semantic Duplicate	Full	0.545	-139.2	-35.57	0.989
	LoRA	0.693	-276.5	-90.21	0.991
No Duplicate	Full	0.850	-103.8	-3.937	0.940
	LoRA	0.650	-252.9	-7.553	0.963



(a) Plots for separability, where `WaterDrum` achieves good separability with high AUROC values.



(b) Plots for calibration, where `WaterDrum` is well calibrated with high R^2 values.

Figure 9: **D1** and **D2** of our `WaterDrum` measured on the Phi-1.5 model for the `WaterDrum-Ax` dataset under the no duplicate and exact duplicate settings.

K. Practical considerations for real-world deployment of `WaterDrum`

K.1. Practical deployment pipeline for `WaterDrum` for evaluating unlearning of LLMs

A key strength of `WaterDrum` is its real-world feasibility, especially when dealing with closed-sourced LLM providers, where other LLM unlearning metrics fail. Unlike other methods, `WaterDrum` can be easily implemented in practice with just additional lightweight data preprocessing and no other changes to existing pipelines. Specifically, `WaterDrum` offers the following advantages for real-world deployment:

- Data owners can quickly watermark their data before sharing them with the model owners or releasing important data publicly. This not only facilitates unlearning verification but also allows them to detect whether their data has been used by model owners without authorization (Lau et al., 2024; Maini et al., 2024).
- No changes are required on the model owners' end. They can continue training their closed-source LLMs and provide API access, or even release open-source models.
- Data owners can then detect whether their data has been used for fine-tuning of any LLM based on just LLM output (even closed-source), submit an unlearning request, and verify whether unlearning has been done via `WaterDrum`. Verification of the `Waterfall` watermark is very efficient (Lau et al., 2024) and can even be run on a CPU (about 3 seconds per 1000 query outputs).
- In comparison, other LLM unlearning metrics face severe limitations that rule out practical deployment, such as requiring a retrained model (D3), which even a cooperative model owner cannot provide due to computational costs.

K.2. Practical real-life use case for D2 (Calibration) in `WaterDrum`

Although it is ideal for unlearning to delete the forget set completely, in practice, partial unlearning of the forget set (as an outcome of imperfect unlearning) may be inevitable due to the size and complexity of LLMs. This is because a) exact unlearning involving retraining from scratch is prohibitively expensive and impractical, and b) perfect unlearning on LLMs is not yet achievable with current approximate unlearning algorithms without significantly harming model performance (e.g., on the retain set).

In App. H.4, we demonstrate this by testing various SOTA unlearning methods: all methods only achieve imperfect unlearning except when the LLM is destroyed (i.e., has no presence of both the forget and retain sets), or when a new LLM is retrained from scratch. With **D2** (Calibration), the characterization of imperfect unlearning becomes possible, and this is important across various stages of the unlearning pipeline in practical, real-life scenarios:

1. Deployment: In practice, model owners may only be able to achieve partial unlearning of the forget set while preserving the utility of their LLM offering to customers. A calibrated continuous score unlearning metric satisfying **D2** such as ours can serve as an objective proxy for negotiations with data owners on the needed extent of unlearning and the corresponding amount of compensation required. The negotiated targeted extent of unlearning can then be used as an objective to guide the actual implementation of unlearning, e.g. the selection of the most suitable unlearning algorithm which may each achieve different forget-retain performance trade-offs (e.g., from a reference plot like Fig. 7, choosing the method that achieves the highest retain score for a fixed forget threshold), or suitable hyperparameters for a given method.

2. Evaluation and development: For research and development, a calibrated metric satisfying **D2** enables evaluation beyond binary success/failure and instead quantifies partial success in unlearning the forget set. This supports a more realistic and granular assessment of theoretical unlearning algorithms.

In summary, perfect unlearning may not be achievable in practice due to the limitations of current LLM unlearning algorithms, which necessitate a continuous evaluation that goes beyond a binary decision. **D2** (Calibration) provides an interpretable way to measure imperfect unlearning, enabling practical evaluation and considerations of trade-offs between LLM performance and compensations. Until perfect unlearning is feasible, a continuous and calibrated metric satisfying **D2** will be valuable.

K.3. Practical real-life scenarios of data owners with similar data

As discussed in Sec. 2, it is common for the data owners to have semantically similar instances, such as news articles on the same event. Here, we identify a real-life scenario where two news agencies, Reuters and The Straits Times (i.e., the data owners), produce semantically similar news articles, as shown in Fig. 10a. These two articles from the two different data owners exhibit a high semantic similarity with an STS score of 0.90. In this case, one agency may request unlearning, which matches our problem setting in **D4**. As another example in the `WaterDrum-Ax` dataset, Fig. 10b shows that the two arXiv paper abstracts from the same *Materials Science* category but different authors (i.e., the data owners) are also semantically similar with an STS score of 0.88. In this example, one group of authors may request unlearning, which also matches our problem setting in **D4**.

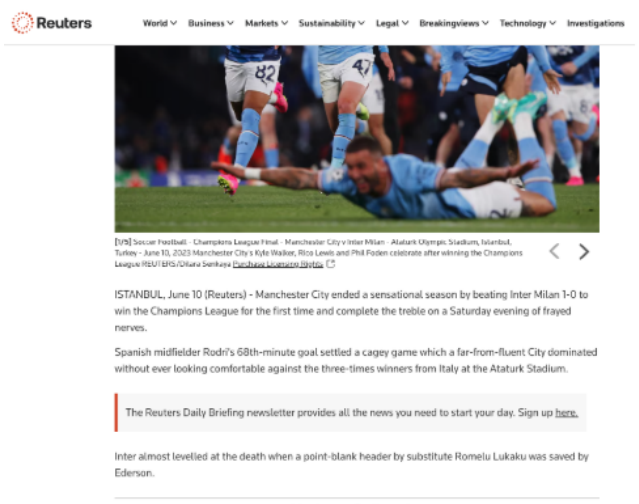
L. Additional Results on Unlearning Evaluation

Here we provide additional evaluation results to the main experiments on both `WaterDrum-Ax` and `WaterDrum-TOFU` datasets.

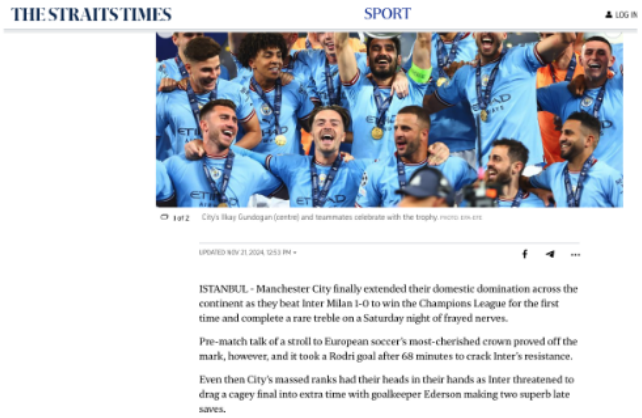
L.1. Evaluation on `WaterDrum-Ax`

L.1.1. Robustness to similar data

Relaxation of Feasibility. In App. H.3, we have demonstrated the calibration of the metrics without access to $\varphi_{\mathcal{R}}$. As illustrated in Fig. 6, the reference metric value varies depending on the similarity between the forget and retain sets. In actual real-world unlearning scenarios, the forget set cannot be known in advance. Only after an unlearning request is made can the reference value be determined with expensive retraining on the actual retain set. This defeats the purpose of using



Martyn Herman, Man City hang tough to beat Inter and complete the treble. Reuter: https://www.reuters.com/sports/soccer/manchester-city-beat-inter-milan-win-champions-league-2023-06-10/. Date of access: Apr 1, 2025



Manchester City beat Inter Milan to win Champions League, clinch treble. The Straits Times: https://str.sg/i3nR. Date of access: Apr 1, 2025.

(a) The news agencies Reuters and The Straits Times both produce news articles reporting on the same soccer match and hence have a high semantic similarity with STS = 0.90.

ABSTRACT

We present the median surface brightness profiles of diffuse Lyα haloes (LAHs) around star-forming galaxies by stacking 155 spectroscopically confirmed Lyα emitters (LAEs) at $3 < z < 4$ in the MUSE Extremely Deep Field (MXDF) with a median Lyα luminosity of $L_{Ly\alpha} \approx 10^{41.1}$erg s^{-1}. After correcting for a systematic surface brightness offset we identified in the data cube, we detect extended Lyα emission out to a distance of ≈270 kpc. The median Lyα surface-brightness profile shows a power-law decrease in the inner 20 kpc and a possible flattening trend at a greater distance. This shape is similar for LAEs with different Lyα luminosities, but the normalisation of the surface-brightness profile increases with luminosity. At distances over 50 kpc, we observe a strong overlap of adjacent LAHs, and the Lyα surface brightness is dominated by the LAHs of nearby LAEs. We find no clear evidence of redshift evolution of the observed Lyα profiles when comparing with samples at $4 < z < 5$ and $5 < z < 6$. Our results are consistent with a scenario in which the inner 20 kpc of the LAH is powered by star formation in the central galaxy, while the LAH beyond a radius of 50 kpc is dominated by photons from surrounding galaxies.

ABSTRACT

The extended Lyα haloes (LAHs) have been found to be prevalent around high-redshift star-forming galaxies. However, the origin of the LAHs is still a subject of debate. Spatially resolved analysis of Lyα profiles provides an important diagnostic. We analyse the average spatial extent and spectral variation of the circumgalactic LAHs by stacking a sample of 155 Lyα emitters (LAEs) at redshift $3 < z < 4$ in the MUSE Extremely Deep Field. Our analysis reveals that, with respect to the Lyα line of the target LAE, the peak of the Lyα line at large distances becomes increasingly more blueshifted up to a projected distance of 60 kpc (≈ 3× virial radius), with a velocity offset of ≈ 250 km/s. This trend is evident in both the mean and median stacks, suggesting that it is a general property of our LAE sample, which typically has a Lyα luminosity ≈ $10^{41.1}$erg s^{-1}. However, due to the absence of systemic redshift data, it remains unclear whether the Lyα line peak at large projected distances is less redshifted compared to the inner regions or truly blueshifted with respect to the systemic velocity. We explore various scenarios to explain the large-scale kinematics of the Lyα line.

(b) In the `WaterDrum-Ax` dataset, the two arXiv paper abstracts from the same *Materials Science* category but different authors both present similar content and hence have a high semantic similarity with STS = 0.88.

Figure 10: Examples of high semantic similarity (STS) across different domains.

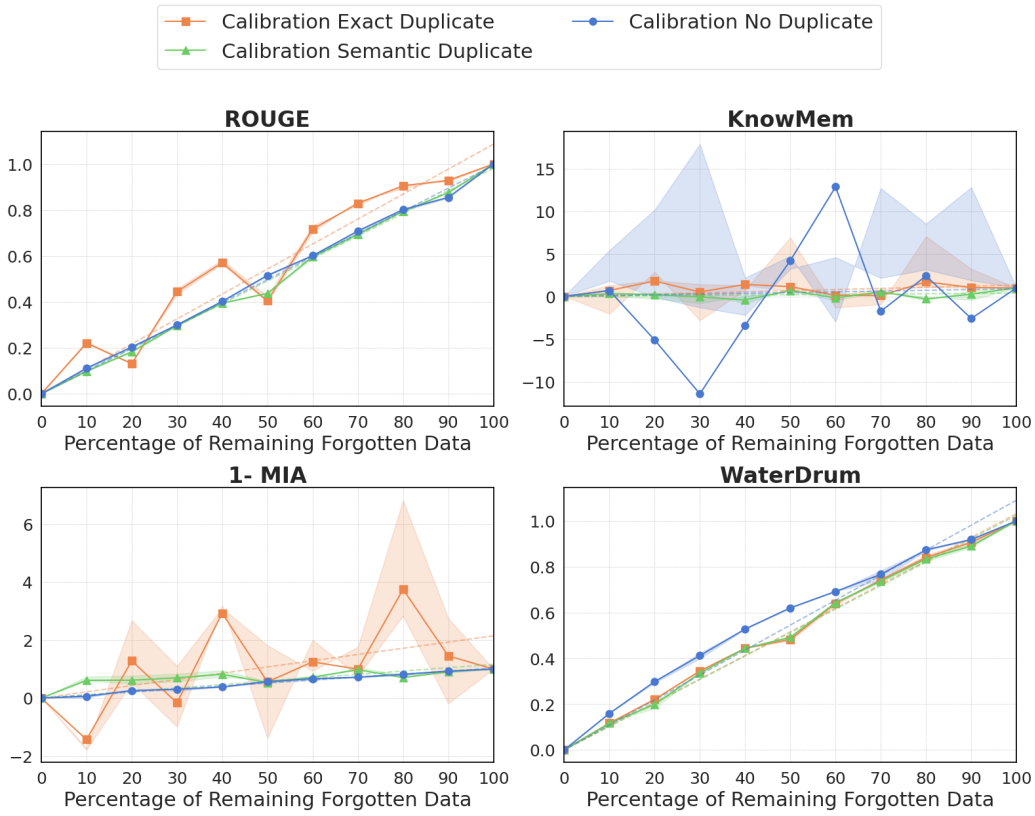


Figure 11: Plots of unlearning metrics against the % of $\mathcal{D}_\mathcal{F}$ remaining in the retrained model, scaled by referencing the original and retrained model with different levels of data similarity for the `WaterDrum-Ax` dataset.

Table 9: R^2 of the best fit line (dotted in Fig. 11 and scaled by referencing the original and retrained model) for various metrics under different levels of similarity for the `WaterDrum-Ax` dataset.

Similarity	ROUGE	KnowMem	MIA	`WaterDrum`
Exact Duplicate	0.923	-0.331	0.273	0.994
Semantic Duplicate	0.997	0.101	-0.011	0.995
No Duplicate	0.998	0.006	0.990	0.957

cheaper (but approximate) unlearning algorithms that avoid retraining. Despite this limitation, here we explore relaxing the restriction by allowing metrics to use $\varphi_\mathcal{R}$ as a reference.

By referencing the fully retrained model as the baseline 0 point for $M(\varphi_\mathcal{R}(q_\mathcal{F}), \mathcal{F})$ (as described in App. F.1), we visualize the scaled calibration of the baseline metrics in Figure 11, and present the R^2 values in Table 9. The results imply that, under the relaxed condition by referencing $\varphi_\mathcal{R}$, the calibration of the baseline metrics generally improves. Notably, ROUGE achieves a good calibration across various similarity levels, though it underperforms in the 'exact duplicate' settings. In contrast, our `WaterDrum` consistently demonstrates strong calibration, with robust R^2 values across all settings. Despite these, it is important to emphasize that the retrained models are not available in practical scenarios, and their availability will eliminate the need to perform unlearning in the first place.

L.2. Evaluations on `WaterDrum-TOFU`

As a supplement to the main experiments, here we present additional results on the `WaterDrum-TOFU` dataset. As described in App. H.1, we consider the exact duplication, semantic duplication, and no duplication settings, and finetune the models on the `WaterDrum-TOFU` dataset. While App. H.2 discusses separability results with similar data, we report here the evaluation of calibration (**D2**) with similar data as follows:

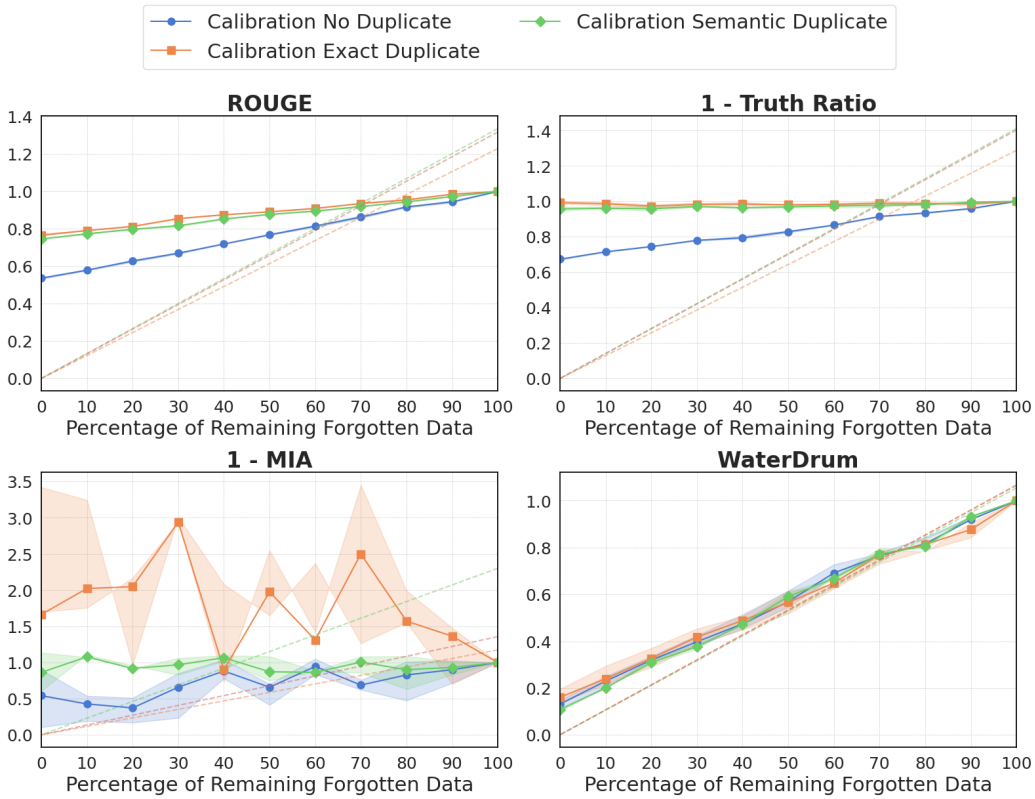


Figure 12: Plots of unlearning metrics against the % of $\mathcal{D}_\mathcal{F}$ remaining in the retrained model, under settings with different levels of data similarity for the `WaterDrum-TOFU` dataset.

Table 10: R^2 of the best fit line for various metrics under different levels of similarity for the `WaterDrum-TOFU` dataset.

Similarity	ROUGE	Truth Ratio	MIA	`WaterDrum`
Exact Duplicate	-30.085	-6444.874	-3.480	0.889
Semantic Duplicate	-24.386	-1416.284	-41.15	0.947
No Duplicate	-2.744	-11.741	-0.838	0.923

L.2.1. Calibration with similar data

Figure 12 visualizes the calibration on `WaterDrum-TOFU` and Table 10 displays the R^2 values. Similar to App. H.3, our `WaterDrum` outperforms the baseline metrics by ensuring $M(\varphi_\mathcal{R}(q_\mathcal{F}), \mathcal{F}) = 0$ and maintaining strong calibration, with high R^2 values without referencing retrained models across all settings.

L.3. Benchmarking unlearning algorithms for more classes and duplicate data

In addition to the results in App. H.4, here we consider the `WaterDrum-Ax` with 1, 3, and 5 data owners (out of 20 total data owners) requesting their data to be unlearned from the LLM (Fig. 14). Additionally, we also consider duplicate data in both forget and retain sets (Fig. 15). We can observe that, except for Finetune, all the other unlearning algorithms perform poorly. However, note that Finetune requires a significant amount of computation resources as the retain set is likely to be significantly larger than the forget set, and almost similar in size to the full dataset. Typically, LLM training only involves very few epochs (Touvron et al., 2023). The computational cost of finetuning a few epochs on the retain set can be almost as expensive as retraining.

We noticed that the retain watermark strength for the retraining model when considering unlearning of 5 classes increases slightly beyond 1.0. We hypothesize that this is due to the large proportion of forget set out of the whole dataset when removing 5 out of the total 20 classes (25% of the training data). The high proportion means that the retain set $\mathcal{D}_\mathcal{R}$ used for

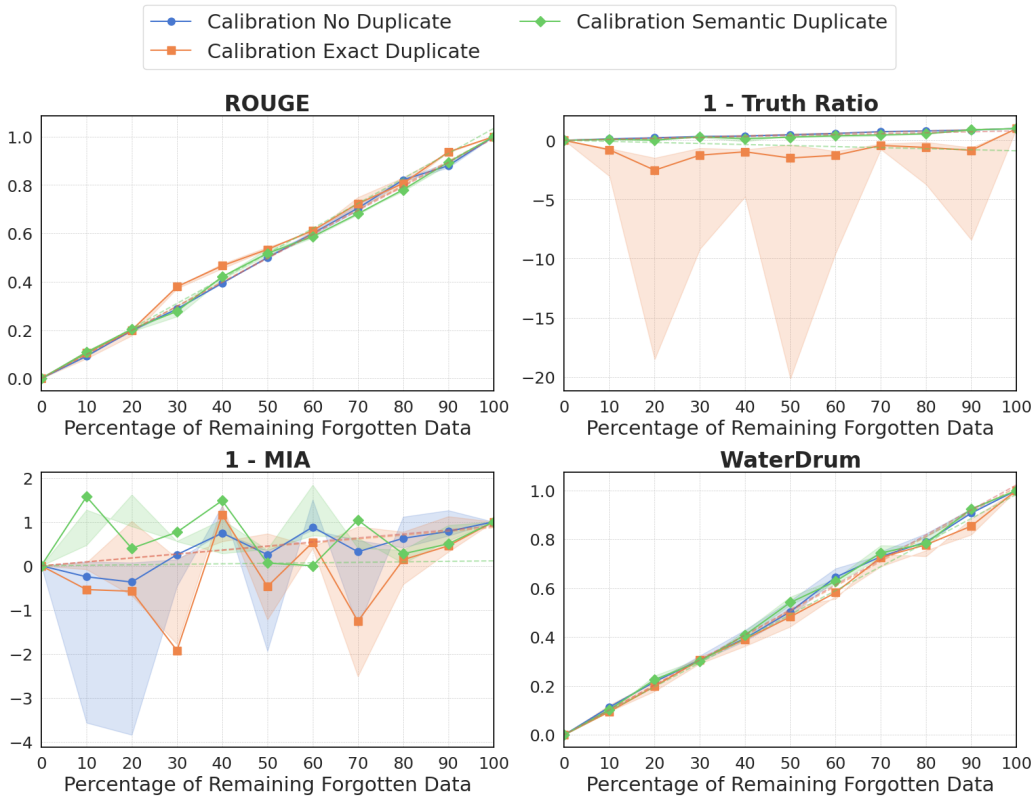


Figure 13: Plots of unlearning metrics against the % of $\mathcal{D}_\mathcal{F}$ remaining in the retrained model, scaled by referencing the original and retrained model with different levels of data similarity for the `WaterDrum-TOFU` dataset.

Table 11: R^2 of the best fit line (scaled by referencing the original and retrained model) for various metrics under different levels of similarity for the `WaterDrum-TOFU` dataset.

Similarity	ROUGE	Truth Ratio	MIA	`WaterDrum`
Exact Duplicate	0.991	-0.586	-0.018	0.997
Semantic Duplicate	0.998	0.854	-0.417	0.996
No Duplicate	0.999	0.995	0.608	0.997

training the retraining model is much smaller than the full dataset $\mathcal{D}_\mathcal{T}$, which could have resulted in the retraining model becoming more specialized in the smaller retraining dataset containing the retain set, resulting in a higher retain watermark strength.

M. Limitations

While our desiderata may be non-exhaustive and watermark strength is just one aspect of unlearning effectiveness, we believe that our work is the first step towards developing more effective and practical unlearning algorithms and deriving theoretical results. Future work could conduct a more comprehensive and systematic evaluation of existing LLM unlearning algorithms and adapt theoretical insights from the watermarking community to analyze LLM unlearning metrics based on our new connection.

N. Other Questions

1. **What is the difference with existing watermarking-based unlearning metric?** Existing watermarking-based unlearning metrics are mostly for image-based classification model, as opposed to our metric for text-based generative LLMs. See discussion on watermark based metrics in App. A for details.

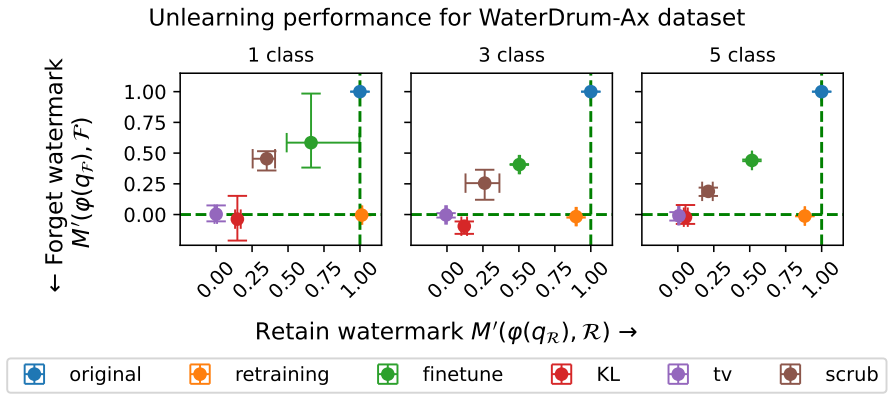


Figure 14: Benchmark of existing unlearning methods with `WaterDrum` on the `WaterDrum-Ax` with no duplication between retain and forget set ($\mathcal{D}_{\mathcal{T}} = \mathcal{D}_{\mathcal{R}} \bigcup \mathcal{D}_{\mathcal{F}}$), for 1, 3, and 5 data owners requesting for their data to be removed.

2. **Existing works (Liu et al., 2025; Lynch et al., 2024) have already identified similar limitations about existing unlearning metrics. What is the novelty of the work?** We formally define clear desiderata and propose a non-retraining based metric that works despite greater similarity between the forget and retain set and the generalization ability of LLMs. See more discussion in App. A.

3. **Why do we only run experiments on TOFU and `WaterDrum-Ax` instead of other datasets such as WMDP?** TOFU and `WaterDrum-Ax` cover both LLM question-answering and generation tasks, which are representative of LLM tasks. WMDP is different from TOFU and `WaterDrum-Ax` in nature because it is specifically for knowledge editing and only contains testing data instead of training data. As our work considers a data-centric view of unlearning, we are concerned with the unlearning of specific data owners' contribution (with potential similar overlapping data across data owners), rather than indiscriminately unlearning certain (harmful) knowledge.

4. **Can our conclusion be generalized to other datasets or other models?** Results on Phi-1.5 (see App. J.2) show that the conclusions can be generalized to other models as well. The two models considered in our paper are representative of recent LLMs, different in terms of model architectural details, and span different model scales. These two models are also the only models considered in (Maini et al., 2024; Wang et al., 2025).

5. **Beyond unlearning effectiveness, can our watermark metric be used to measure utility preservation/retention?** As shown in App. H.4, our metric can be used to verify that the metric on the retain set in the unlearned model is similar to that in the original model. Hence, by verifying the retain watermark, our metric can also quantify the extent of undesirable removal of the retain set's influence and evaluate the effects of catastrophic forgetting.

6. **Practical significance of unlearning from finetuning data vs pretraining data.** In real-life applications, LLM finetuning is performed to enhance the model in specific downstream tasks, which is more likely to make use of task-specific datasets. These datasets are more concerned with privacy/safety issues, and are hence more significant for unlearning than public datasets.

7. **What new insights can be gained from the proposed framework? (a)** We showed that existing metrics fail on our necessary desiderata (App. F.1), prompting caution on metrics design. **(b)** Using `WaterDrum` to benchmark LLM unlearning algorithms (App. H.4) shows that they perform poorly on unlearning and retaining performance. `WaterDrum` can serve as an optimization criterion for future LLM unlearning algorithms. **(c)** By emphasizing practical conditions, `WaterDrum` encourages future LLM unlearning algorithms to consider realistic constraints.

8. **Why do we not consider other desiderata?** Our work focuses on the most essential desiderata (effectiveness desiderata) and more practical/realistic settings. These desiderata are those that we find to be most relevant necessary criteria for effective unlearning metrics, though they are not meant to be exhaustive nor by themselves sufficient to guarantee unlearning. We see our work as complementary to other compatible frameworks.

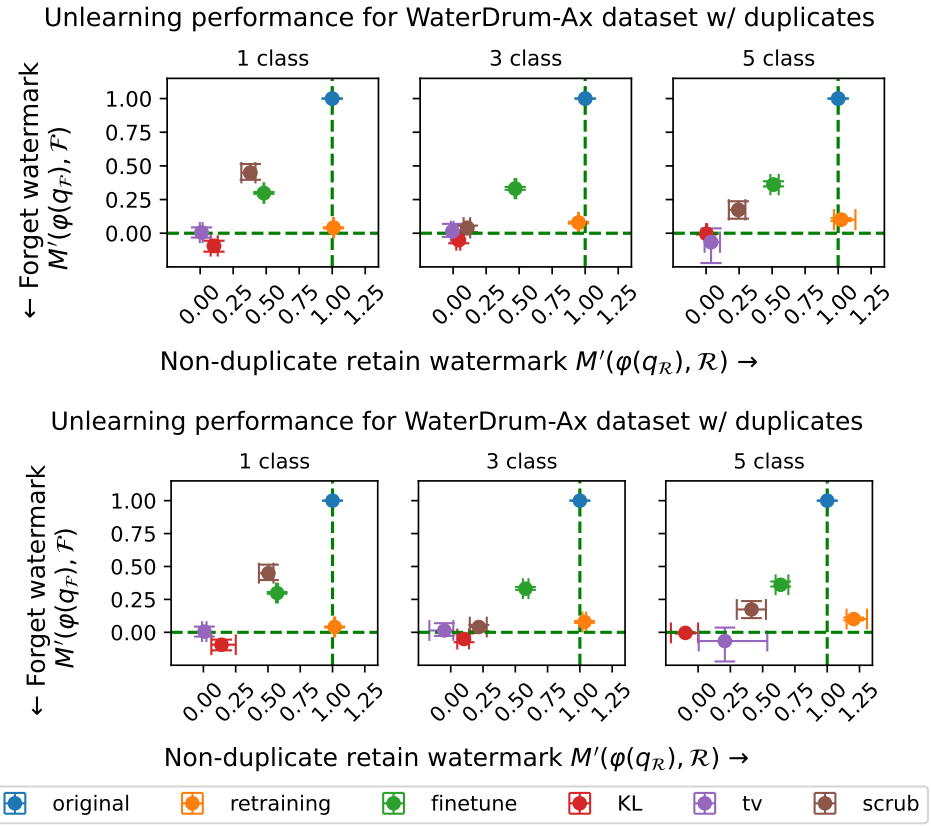


Figure 15: Benchmark of existing unlearning methods with `WaterDrum` on the `WaterDrum-Ax` with duplicate data ($\mathcal{D}_{\mathcal{T}} = \mathcal{D}_{\mathcal{R}} \bigcup \mathcal{D}_{\mathcal{F}} \bigcup \mathcal{D}_{\mathcal{S}}$, where $\mathcal{D}_{\mathcal{F}}$ and $\mathcal{D}_{\mathcal{S}}$ are the duplicate data in the forget and retain sets respectively). For the x-axis, the top figures show `WaterDrum` scores for the retain set excluding duplicates $\mathcal{D}_{\mathcal{R}}$, while the bottom figure shows `WaterDrum` scores for only the duplicates within the retain set $\mathcal{D}_{\mathcal{S}}$. The y-axis for both figures are the same, showing $\mathcal{D}_{\mathcal{F}}$.