Rectifying Privacy and Efficacy Measurements in Machine Unlearning: A New Inference Attack Perspective

Nima Naderloui[1], Shenao Yan[1], Binghui Wang[2], Jie Fu[3], Wendy Hui Wang[3], Weiran Liu[4], Yuan Hong[1]
[1]*University of Connecticut,* [2]*Illinois Institute of Technology,*
[3]*Stevens Institute of Technology,* [4]*Alibaba Group*

Abstract

Machine unlearning focuses on efficiently removing specific data from trained models, addressing privacy and compliance concerns with reasonable costs. Although exact unlearning ensures complete data removal equivalent to retraining, it is impractical for large-scale models, leading to growing interest in inexact unlearning methods. However, the lack of formal guarantees in these methods necessitates the need for robust evaluation frameworks to assess their privacy and effectiveness. In this work, we first identify several key pitfalls of the existing unlearning evaluation frameworks, e.g., focusing on average-case evaluation or targeting random samples for evaluation, incomplete comparisons with the retraining baseline. Then, we propose RULI (*Rectified Unlearning Evaluation Framework via Likelihood Inference*), a novel framework to address critical gaps in the evaluation of inexact unlearning methods. RULI introduces a dual-objective attack to measure both unlearning efficacy and privacy risks at a per-sample granularity. Our findings reveal significant vulnerabilities in state-of-the-art unlearning methods, where RULI achieves higher attack success rates, exposing privacy risks underestimated by existing methods. Built on a game-based foundation and validated through empirical evaluations on both image and text data (*spanning tasks from classification to generation*), RULI provides a rigorous, scalable, and fine-grained methodology for evaluating unlearning techniques.

1 Introduction

Unlearning is crucial in modern machine learning, enabling the efficient removal of specific data (unlearn samples) or knowledge from trained models. It ensures compliance with privacy laws [1, 11], corrects outdated [32] or harmful content [7], and keeps models ethical and adaptable. As models and datasets continue to scale, interest in machine unlearning [9, 37, 70] has grown rapidly. While retraining from scratch without the removed samples provides a theoretically sound solution, it remains computationally expensive. To address this, a variety of inexact unlearning methods [9, 36] have been proposed as more efficient alternatives. Recent efforts focus on improving the efficiency, robustness, and accuracy of these approaches. Accordingly, rigorous evaluation is critical to ensure their practical effectiveness and to support trustworthy deployment by model providers and users alike.

First, from a privacy perspective, techniques such as differentially private training [2] and defenses against inference attacks [27, 53, 68] typically require strong empirical validation—often through *inference attacks* (e.g., membership inference attacks [3, 52, 55])—to assess their effectiveness in mitigating privacy risks under worst-case or average-case scenarios. Substantial research efforts have been dedicated to the design and evaluation of privacy mechanisms, often supported by empirical studies using inference attacks.

Inexact unlearning can also be considered as a post-hoc privacy mechanism to efficiently remove samples and therefore, selectively [34] protect the privacy for a portion of data upon requests. Similarly, we require strong verifications to evaluate and minimize the privacy leakage for any unlearned sample [38, 42]. Existing unlearning works often report high protection against privacy attacks [18, 44, 79]. However, Hayes et al. initiated efforts to advance stronger sample-level attacks [38] indicating stronger attackers are required to evaluate unlearning. According to [38], existing algorithms are more vulnerable to their per-sample [12] adapted attack. These insights motivate us to further investigate the following questions.

*Q1 [**Privacy**]: Can existing inference attacks accurately measure the privacy leakage from unlearned models?*

Second, another key requirement for evaluating unlearning success is assessing how closely it approximates the gold standard of retraining [38, 76].

*Q2 [**Efficacy**]: How to accurately measure the difference between the unlearned model and the retrained model?*

Since efficacy reflects whether the requested samples have been effectively removed from the model through unlearning, it is related to (but distinct from) the privacy of the unlearned samples. While privacy concerns the risk of information leakage (“existence of unlearned samples”), efficacy focuses on the actual removal of the samples from the model.

1.1 Contributions

Motivated by the above questions, this work contributes the following to the field of machine unlearning.

(1) Identifying the limitations of existing inference attacks in evaluating unlearning. We observe that current inference attacks on unlearning methods [26, 38, 46] might not sufficiently challenge unlearning algorithms to enable a comprehensive evaluation of their effectiveness. First, most evaluations typically focus on *average-case* scenarios and targeting *random* samples, neglecting the *per-sample* vulnerabilities associated with specific high-risk (or particularly vulnerable) data samples [13]. Second, we also observe that simply comparing the retrained model with the unlearned model's accuracy may not yield an accurate assessment of privacy leakage [38].

(2) Developing novel inference attacks for dual measurements on privacy leakage and efficacy. To address these deficiencies, we revisit the theoretical foundations of adversarial settings and design novel membership inference attacks (MIAs) [12, 19, 62] with two key objectives:

- *Privacy Leakage*: an MIA to assess whether a sample has been unlearned or was never part of the training set (and subsequently not unlearned) based solely on inferring the unlearned model.
- *Efficacy*: an MIA to assess whether a sample's inference output corresponds to an unlearned or retrained model.

With new MIAs, we propose a novel unlearning evaluation framework RULI (*Rectified Unlearning Evaluation Framework via Likelihood Inference*). To our best knowledge, RULI takes the first step to perform per-sample, targeted attacks on vulnerable samples in unlearning using refined membership signals and hypothesis testing, enabling the evaluation of both efficacy and privacy leakage at a reasonable attack cost.

(3) High attack performance and accurate unlearning evaluation. We evaluate RULI against state-of-the-art (SOTA) attacks on standard unlearning benchmarks and conduct a comprehensive study of how different target samples exhibit varying levels of privacy leakage. For example, using a Gradient Ascent-based unlearning method, RULI achieves at least 20% TPR@1% FPR on CIFAR-10 and CIFAR-100, and up to 54% TPR@1% FPR when unlearning targeted 7-gram sequences from the WikiText-103 dataset. Our results also demonstrate that existing average-case attacks [26, 38, 46] and advanced methods like U-LiRA [38] have substantially underestimated the privacy leakage (and efficacy) especially under target random samples.

As Figure 1 shows, we uncover significantly stronger privacy leakage and highlight that efficacy is a distinct metric from privacy. We find that most inexact unlearning methods cannot closely approximate retraining. In other words, inexact unlearning may fail to remove samples as effectively as retraining, and even strong unlearning methods may fail to protect vulnerable samples when injected as canaries.

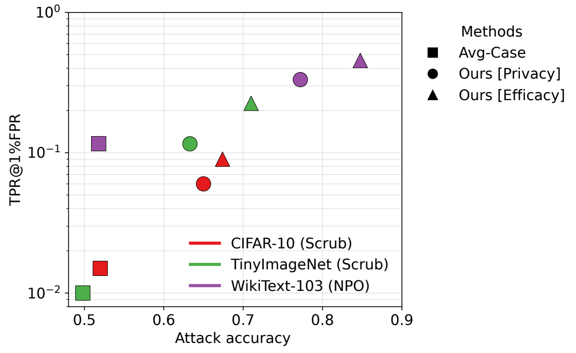


Figure 1: RULI (ours) separates privacy leakage and unlearning efficacy through targeted membership inference. Population average-case attacks [35] consistently *underestimate privacy leakage* and *fail to capture unlearning efficacy*. This figure demonstrates some representative results, e.g., Scrub [44] on CIFAR-10 (ResNet-18) and TinyImageNet (Vision Transformer [47]) for image sample unlearning, and NPO [82] on WikiText-103 (GPT2-small [57]) for token sequence unlearning. U-LiRA [38] falls between RULI and average-case attacks, as it targets random samples on a per-sample basis but considers the efficacy similar to the privacy leakage.

(4) Generalizability. RULI can be generalized to broadly support diverse tasks (from image classification to text generation) in multiple domains. We have validated its effectiveness on both image datasets (e.g., CIFAR-10/100, TinyImageNet) and text datasets (e.g., WikiText-103) by unlearning targeted data samples from various models, such as images from ViT models, and token sequences from language models.

2 Preliminaries

2.1 Machine Unlearning

Machine Unlearning refers to the process of selectively forgetting specific pieces of data that a model has previously trained on. Consider a model, denoted as θ_I, trained on a dataset D_{train}; any samples belonging to D_{train} can be subjected to unlearning, often due to the *right to be forgotten* (Article 17 of the GDPR) [1, 11] and form a forget set D_f. The challenge is to erase the influence of D_f while preserving the utility of the remain data $D_r = D_{\text{train}} \setminus D_f$. Machine unlearning employs various techniques to efficiently remove the influence of D_f from a model originally trained on D_{train}. The objective is for the resulting unlearned model to closely approximate a model trained exclusively on D_r, without requiring full retraining.

Unlearning Measurement. Most inexact unlearning methods [26, 46] consider accuracy measurements to find how much the accuracy on D_r, D_f, and also unseen test data (D_{test}) differs with retraining. Previous works have also explored the use of Membership Inference Attacks (MIAs) [16, 17, 46, 65, 78] to evaluate unlearning, defining success based on the attacker's ability to distinguish whether a sample was unlearned or included in retraining. Unlearning measurement is specifically critical in inexact unlearning algorithms as these methods, despite being efficient and scalable, often do not provide formal guarantees on data removal [36]. Therefore, requires robust evaluations (often by strong attacks [38]) to empirically ensure the unlearning algorithm is comparable to a retrained model and ensures privacy.

2.2 Inexact Unlearning Criteria

Apart from the first goal of unlearning to speed up the data removal process compared to retraining (efficiency), given an initially trained model θ_I and an unlearning algorithm $\mathcal{U}$, the goal of inexact unlearning is to remove a subset $D_f \subset D_{\text{train}}$ from the training data by meeting the following key criteria:

(1) Accuracy. The unlearned model ($\theta_{\mathcal{U}}$) should ideally behave similarly to a retrained model ($\theta_{\mathcal{R}}$), recalling that the retrained model is the model trained excluding the D_f. As mentioned, the basic accuracy test is that $\text{Acc}(\theta_{\mathcal{U}}) \approx \text{Acc}(\theta_{\mathcal{R}})$ on all samples from D_f, D_r, and test data.

(2) Efficacy. A widely accepted definition of the goal of unlearning is to ensure that the distribution of an unlearned model becomes indistinguishable from the distribution of a retrained model, which is trained exclusively on the remained dataset (D_r) [32, 60]. As highlighted in [38], there is a need to develop a tool to empirically estimate this indistinguishability.

(3) Privacy. It focuses on ensuring that the unlearned model ($\theta_{\mathcal{U}}$) provides no residual information about the forget data, D_f. In this context, the objective is that an MIA on D_f using $\theta_{\mathcal{U}}$ should fail to infer whether any sample from D_f was part of the original training process and unlearned [42]. This test emphasizes the privacy protection provided by the unlearning process, ensuring that no attacker can exploit the unlearned model to extract information about any forget sample.

Unlearning vs. MIA-resilience. Unlike MIA-mitigation methods [15, 53, 54, 68], which aim to protect the entire training set, unlearning provides *selective* privacy on demand. It targets only the requested forget set D_f, leaving D_r unaffected unless specified. While applying MIA-resilient training to D_f may reduce privacy leakage, it fails to meet the efficacy requirements. This calls for efficacy measurements besides the privacy leakage to evaluate unlearning.

3 Unlearning Framework and Threat Model

Similar to previous works (e.g., [16, 36, 44, 46, 70]), the framework is divided into two phases: *training* and *unlearning*. In the *training* phase, the model is trained without anticipating future *unlearning requests*, without training assumptions [20, 37] or incorporating additional checkpoints [9, 44]. When an *unlearning request* is received, the unlearning phase is activated as a separate process, efficiently updating the model to produce $\theta_{\mathcal{U}}$ with minimal computational costs. Our approach adheres to this flexible framework, enabling seamless integration of nearly all "inexact unlearning" methods.

Specifically, our unlearning framework involves three main entities: (1) Users (whose data are used for training the model and who may request to have their data unlearned), (2) Model Trainer (who trains the model and executes the unlearning process), and (3) Model Recipient (who can interact with the model, potentially performing inference attacks to evaluate privacy risks [17, 20, 31, 39, 59]).

Users. Individuals or entities who contribute to the data used in the model training. In the unlearning setting, users may request the removal or modification of their data from the trained model, necessitating mechanisms to ensure that their data can be effectively unlearned. These requests can pertain to any type of data contributed during training. Henceforth, we will refer to the specific data contributor and entities who request unlearning as the *users* for simplicity.

Model Trainer. The entity is responsible for storing data, training the model, and executing unlearning. The model trainer ensures that data are securely stored and processes requested queries. In the context of unlearning, it must also handle data removal requests and update the model to exclude the specified data while maintaining model performance and respecting user privacy rights [1, 11]. We consider an honest model trainer, who performs training and unlearning with no adversarial objective against users' privacy.

Attacker's Capabilities. We consider an attacker with black-box access to *only* the final unlearned model. That is, the attacker can query the unlearned model but has no access to the original (pre-unlearning) model or to any intermediate model states during the unlearning process. This assumption differs from some prior works [17, 39], which assume the attacker has access to both the original and unlearned models. In contrast, our threat model aligns with a more realistic deployment scenario: once unlearning is complete, only the final unlearned model is released. This assumption also reflects the nature of inexact unlearning methods, which directly modify the original trained model rather than retraining from scratch. Unlike retraining, which inherently produces intermediate models, inexact unlearning does not expose meaningful unlearning checkpoints throughout the process. Therefore, it is reasonable to assume that the unlearned model becomes available only after the unlearning process is fully completed.

Moreover, following [12, 38, 62], we assume the attacker has knowledge of the training and unlearning algorithms, and can therefore train and access *shadow models* to support the inference attack.

Unlearning Efficacy Evaluator Capabilities. We treat the efficacy evaluator as an honest model trainer who empirically assesses how well unlearning approximates retraining. As shown in Section 4.3.2, querying only the final unlearned model is insufficient for this purpose. Unlike an attacker, the evaluator audits unlearning fidelity, not privacy leakage. The evaluator interacts with models via black-box queries, avoiding parametric comparisons [34, 72], since inexact unlearning does not follow the same gradient path as retraining or guarantees indistinguishability. Instead, we assess behavioral differences resulting from the removal of requested samples.

4 A New Inference Attack on Unlearning

In this section, we examine privacy leakage and efficacy in unlearning by identifying pitfalls in existing methods and introducing a new inference attack (Sections 4.1 and 4.2).

4.1 Pitfalls in Existing Unlearning Evaluation

Unlearning privacy is measured by the attacker's inability to distinguish between unlearned and never-trained samples [38, 42, 44, 61]. This critical problem can be formalized through a game-based framework [13, 49, 59, 78], which simulates the interaction between a challenger, responsible for training and unlearning a model, and an adversary, who aims to infer membership. In the challenger-adversary framework for unlearning, the process begins with a **challenger** C, who trains a model θ_I using a training dataset sampled from data distribution $\mathcal{D}$ ($D_{\text{train}} \subseteq \mathcal{D}$) via a training algorithm $\mathcal{A}$. After initial training, the challenger selectively unlearns a subset of data points, $D_f \subset D_{\text{train}}$, by applying an unlearning algorithm $\mathcal{U}$. The resulting model, after unlearning, is denoted as $\theta_{\mathcal{U}}$. The adversary's (A) attack settings are captured through different games, characterized by different types of access and tools. In this paper, these games can serve as the theoretical representation for evaluating the privacy leakage and efficacy of unlearning.[1]

Game 1: Existing MIA for unlearning privacy

1. The *challenger* trains a model with $D_{\text{train}} \subseteq \mathcal{D}$ and gets θ_I.
2. The *challenger* unlearns $D_f \subset D_{\text{train}}$ to get the unlearned model $\theta_{\mathcal{U}}$.
3. The *challenger* flips a coin c:
   - If $c =$ head, the challenger chooses a data point z from D_f
   - If $c =$ tail, the challenger chooses a data point z from $\mathcal{D} \backslash D_{\text{train}}$
4. The *challenger* sends the selected data point z to the adversary.
5. Given the unlearned model $\theta_{\mathcal{U}}$, the *adversary* queries z to determine if it is in D_{train} and guess $\hat{c} = \{\text{head}, \text{tail}\}$; *adversary* wins if $\hat{c} = c$.

[1]Similar to the instantiated MIAs, the adversary in the game is also capable of training and accessing the shadow models (this applies to all the games and inference attacks defined in this paper).

Game 1 formalizes the existing MIAs for unlearning privacy. Specifically, it assesses the adversary's ability to distinguish between data points that have been unlearned and those that were never part of the training dataset. In this setup, the adversary has neither prior knowledge of nor control over the data samples requested for unlearning. This game simulates a scenario where the adversary can only query *random samples* provided by the challenger, serving as a baseline for measuring privacy leakage in unlearning under non-targeted attacks. However, existing evaluations (mostly built upon this game and the corresponding MIA) exhibit significant shortcomings.

Pitfall I: Average-case MIAs Cannot Fully Disclose Unlearning Privacy. Nearly all existing works [25, 26, 44, 46] rely on MIAs based on **Game 1** to evaluate the privacy leakage of their unlearning methods. The most naive approach [26, 46] involves training an MI-classifier on equal-sized training/testing datasets to measure the attacker's ability to identify an unlearned sample as part of the training. A refined approach [35] employs population attacks, where many shadow models are trained and unlearned, and an MI-classifier is trained on *unlearn* and *out* (excluded from training) populations for distinguishing these two cases. While this improves upon naive approaches, it still evaluates privacy leakage in terms of aggregate metrics, failing to consider the unique memorization status of individual samples. Since memorization varies per sample, unlearning should also be evaluated at the per-sample level for accurate privacy assessment.

Pitfall II: Evaluating *Random* Samples Underestimates Unlearning Privacy. Recent trends in machine learning privacy research emphasize identifying and evaluating vulnerable samples, rather than assessing the privacy of the entire dataset [3, 4]. However, to our best knowledge, nearly all unlearning methods only consider the case of *random sample evaluation* for unlearning—choosing random samples uniformly across all classes or within a single class—without accounting for the specific memorization level of each sample. While recent works [38, 44] suggest that unlearning random samples within a single class poses the greatest challenge, even more extreme cases can further challenge unlearning. More importantly, we show that evaluating random samples tends to underestimate the privacy leakage in unlearning.

Pitfall III: Incomplete Comparisons with the *Retrain* Baseline (Efficacy). Many inexact unlearning methods are compared to a retrained model [74] to ensure that the unlearned model behaves similarly to it. However, these comparisons are often limited to accuracy metrics [16, 26, 44], which might be inaccurate (see Table 2 and Figure 5). For example, a model showing close accuracy performance to the retrained model might unlearn data samples differently than the retrained model. We emphasize the need to evaluate how individual samples are unlearned relative to their retrained counterparts. While this pitfall does not directly relate to privacy, it highlights the limitations of existing methods in quantifying the similarity between unlearned and retrained models beyond simple accuracy comparisons.

4.2 Avoiding Pitfalls Requires New Games

Previous studies have assessed the unlearning success by observing the attacker's inability to execute an inference attack [42, 44] on the unlearned model. As discussed in **Pitfall I**, although many unlearning benchmarks have demonstrated *average-case* MIA-resilience, our focus is on evaluating unlearning through sample-specific membership signals for identifying underestimated privacy leakage, in line with recent privacy evaluation literature [3, 12, 52, 80].

Per-Sample Privacy Evaluation for Unlearning. MIAs are typically evaluated using metrics such as the Area Under the Curve (AUC) and TPR@lowFPR over the entire training dataset. For example, in CIFAR-10, the target training data consists of 50,000 samples equally split between trained and not-trained samples [12] However, MIA performance and memorization behavior are not uniform across all samples [3, 12]. As shown in Table 10 (in the experiments), targeting smaller, randomly selected subsets of training data often leads to degraded attack performance [3], likely due to the increased likelihood of these smaller subsets containing predominantly safe samples [13]. Consequently, uniformly selected smaller subsets result in lower attack accuracy and reduced TPR@lowFPR. This challenge is particularly relevant in the context of unlearning. Similarly, the target set (i.e., D_{target}) of membership inference is typically a smaller fraction of the training set, often less than 10% [9]. Additionally, most unlearning algorithms are not designed to handle the removal of large portions of data [26]. Thus, evaluations should not only consider individual samples but also include challenging scenarios for unlearning algorithms, avoiding **Pitfall II**.

Game 2: Targeted MIA for unlearning privacy

1. The *challenger* trains a model with $D_{\text{train}} \subseteq \mathcal{D}$ and gets $\theta_{\mathcal{I}}$.
2. The *adversary* chooses a target set D_{target} and sends to *challenger*.
3. The *challenger* unlearns $D_f \cup \{D_{\text{train}} \cap D_{\text{target}}\}$ to get the model $\theta_{\mathcal{U}}$.
4. The *challenger* flips a coin c:
   - If $c =$ head, the challenger chooses a data point z from $D_f \cap D_{\text{target}}$
   - If $c =$ tail, the challenger chooses a data point z from $D_{\text{target}} \setminus D_{\text{train}}$
5. The *challenger* sends the selected data point z to the adversary.
6. Given the unlearned model $\theta_{\mathcal{U}}$, the *adversary* queries z to determine if it is in D_{train} and guess $\hat{c}$ = {head, tail}; *adversary* wins if $\hat{c} = c$.

Targeted MIA for Evaluating Unlearning Privacy. To address these challenges, our approach is introduced in **Game 2** (new contents marked in blue), which extends the attack setting to allow targeted access to the unlearned model. In this setting, the adversary selects specific targeted samples D_{target} for unlearning rather than relying on random samples. Despite this extended capability, the adversary operates in a *targeted black-box* threat model [73, 75], lacking access to internal parameters, gradients, or embeddings and unable to manipulate the unlearning process directly.

Game 3: MIA for unlearning efficacy

1. The *challenger* trains a model with $D_{\text{train}} \subseteq \mathcal{D}$ and gets $\theta_{\mathcal{I}}$.
2. The *adversary* chooses a target set D_{target} and sends to *challenger*.
3. The challenger unlearns $D_f \cup \{D_{\text{train}} \cap D_{\text{target}}\}$ to get the model $\theta_{\mathcal{U}}$.
4. The challenger flips a coin c:
   - If $c =$ head, the challenger chooses a data point z from $D_f \cap D_{\text{target}}$, and the query result will be given as $f_{\theta_{\mathcal{U}}}(\cdot)$
   - If $c =$ tail, the challenger chooses a data point z from $D_{\text{target}} \setminus D_{\text{train}}$, and the query result will be given as $f_{\theta_{\mathcal{I}}}(\cdot)$
5. The *challenger* sends the selected data point z to the adversary.
6. Given the query from queries z as $f_\theta(\cdot)$, the *adversary* determines if z is in D_f and guess $\hat{c}$ = {head, tail}; *adversary* wins if $\hat{c} = c$.

Evaluating Efficacy with Indistinguishability between Unlearned Model and Retrained Model. As highlighted in **Pitfall III**, inexact unlearning lacks formal guarantees, making empirical evaluation essential [38]. To assess whether an unlearning method approximates retraining behavior, we adopt a formal game formulation based on indistinguishability between the unlearned model and a retrained model.

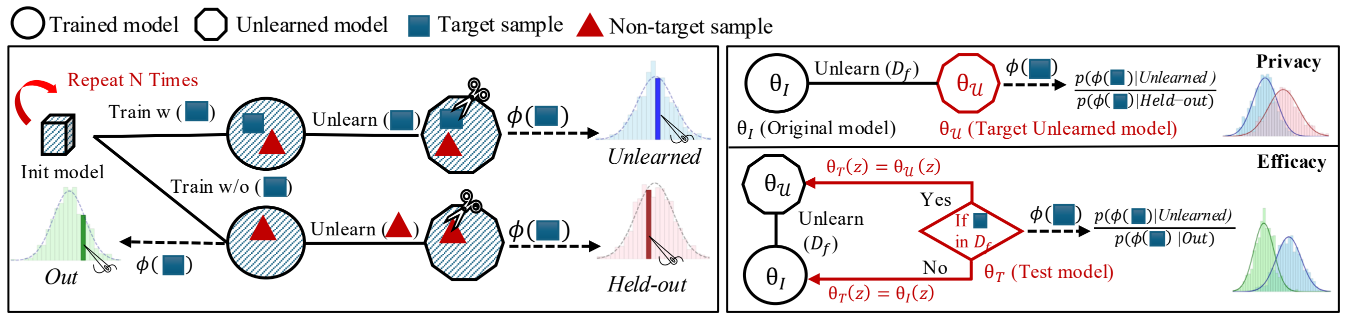
In **Game 3** (new contents marked in pink), the challenger first trains and unlearns a model to obtain $\theta_{\mathcal{U}}$. The challenger flips a coin and chooses the target sample from the target set chosen and shared by the adversary. If the sample belongs to D_f, the query result will be given to the adversary as $f_\theta(\cdot) = f_{\theta_{\mathcal{U}}}(\cdot)$; and if the sample was not involved in training, the query result will be given as $f_\theta(\cdot) = f_{\theta_I}(\cdot)$. The adversary must guess if that sample is unlearned or retrained. *This game is fair:* the adversary receives output from only one model and never has access to internal weights or both models simultaneously. The baseline success rate remains 50% (random guessing), and the adversary gains no extra advantage. The game only tests distinguishability between 1) a sample that was unlearned, and 2) a sample that was never trained on, based solely on observable predictions.

We highlight that **Game 2** is a practical privacy attack for identifying the privacy leakage in the unlearned model $\theta_{\mathcal{U}}$. **Game 3** defines the foundation for an attack to differentiate the retrained model from an unlearned model (efficacy), and it is not instantiated as a real-world attack.

4.3 RULI to Resolve Pitfalls

4.3.1 Theoretical Foundation for RULI

Earlier, we have established the theoretical foundation to tackle Pitfalls. Now we design a new MIA for instantiating the attack from the empirical perspective. Similar to [12], we



(a) Shadow model training and unlearning

(b) Privacy and efficacy measurements

Figure 2: Overview of RULI shadow model training and unlearning to measure the privacy leakage and efficacy.

begin with the distributions an adversary requires to make a hypothesis test [12] on them. According to **Game 2**, adversary requires two distinct distributions of unlearned models.

$$Q_u = \{\theta \leftarrow \mathcal{U}(\mathcal{A}(D_{\text{train}} \cup \{z\}), \{z\} \cup D_f \mid D_f \subset D_{\text{train}} \sim \mathcal{D})\}$$
$$Q_h = \{\theta \leftarrow \mathcal{U}(\mathcal{A}(D_{\text{train}}), D_f \mid D_f \subset D_{\text{train}} \sim \mathcal{D})\}$$

where $\mathcal{U}$ and $\mathcal{A}$ refer to the unlearning and training algorithms, respectively. Specifically, Q_u denotes the distribution of models in which the target sample (z) was included in training and subsequently unlearned, while Q_h represents the distribution of models where $\{z\}$ was never part of the training set and thus not unlearned. Hence, given an unlearned model $\theta_{\mathcal{U}}$, the attacker makes a likelihood ratio test [12] to derive the membership of the target sample according to Equation 1.

$$\Lambda(z) = \frac{p(\theta_{\mathcal{U}} \mid Q_u(z))}{p(\theta_{\mathcal{U}} \mid Q_h(z))}, \quad \forall(z) \in D_{\text{target}} \tag{1}$$

Furthermore, regarding **Game 3**, attacker essentially requires these distributions:

$$Q_u = \{\theta \leftarrow \mathcal{U}(\mathcal{A}(D_{\text{train}} \cup \{z\}), \{z\} \cup D_f \mid D_f \subset D_{\text{train}} \sim \mathcal{D})\}$$
$$Q_R = \{\theta \leftarrow \mathcal{A}(D_{\text{train}} \mid D_{\text{train}} \sim \mathcal{D})\}$$

Similarly, Q_R represents the distribution of the models trained without the target sample (z) ("*retrained excluding* $\{z\}$"). Different from previous tests, the hypothesis test here cannot be on a single unlearned model or retrained model. Hence, we introduce a new *unlearning test model* as:

$$\theta_{\mathcal{T}}(z) = \begin{cases} \theta_{\mathcal{U}}(z) & \text{if } z \in D_{\text{target}} \cap D_{\text{train}} \\ \theta_{\mathcal{R}}(z) & \text{if } z \in D_{\text{target}} \setminus D_{\text{train}} \end{cases} \tag{2}$$

Then, given a test model $\theta_{\mathcal{T}}$, the following likelihood ratio test can be adopted to derive the efficacy based on **Game 3**.

$$\Psi(z) = \frac{p(\theta_{\mathcal{T}} \mid Q_u(z))}{p(\theta_{\mathcal{T}} \mid Q_R(z))}, \quad \forall(z) \in D_{\text{target}} \tag{3}$$

In practice, the test model is introduced to enable per-sample efficacy evaluation without requiring direct access to both the unlearned and retrained models for comparison. Since each target sample may either be unlearned or never trained on, but we only have access to individual models trained under one configuration at a time, we use $\theta_{\mathcal{T}}(z)$ as a unified interface. It returns the output from the correct model depending on $\{z\}$'s status: unlearned or excluded. Figure 2 (b) shows how a test model operates in RULI. If a target sample is unlearned, a query would be given from the unlearned model. Otherwise, the query would be given from the model that is trained, excluding the target sample, and the efficacy would be evaluated from a likelihood test per Equation 3.

4.3.2 MIA in RULI

Membership Inference in RULI. Algorithm 1 outlines the attack pipeline in RULI to evaluate the privacy leakage and unlearning efficacy. The attacker trains N shadow models using samples drawn from the data distribution $\mathcal{D}$. For each target sample z, the algorithm ensures that model outputs (observations) are collected under three training scenarios: when z is included in training (*In*), when it is excluded entirely (*Out*), and when it is included and later unlearned via a known unlearning algorithm (*Unlearned*). It also simulates when it is *Out* and the model is unlearned (*Held-out*). After each model is trained, the inference function ϕ is applied to obtain the model's observation on the target sample, and the result is stored in one of five observation sets. Figure 2 shows an example for the shadow model training and unlearning.

In the second step, Kernel Density Estimation (KDE) [64] is applied to fit smooth distributions over the collected confidence scores from each distribution. The attacker then queries the actual unlearned model and computes two likelihood ratio tests: Λ, which compares the unlearned distribution to the held-out distribution to evaluate *privacy leakage*, and Ψ, which compares the unlearned distribution to the out distribution (excluded samples) to measure the unlearning *efficacy*.

Parallelizing MIA to All Target Samples. To scale the algorithm to a set of target samples while maintaining balanced per-sample coverage across roles, we structure the shadow training in groups of three. In each iteration, we select a disjoint subset of target samples of size $\frac{N}{3}$, and assign each sample to a unique role within that batch: one-third as D_{in}, one-third as D_{unlearn}, and one-third as D_{out}. We train shadow models $D_{\text{in}} \cup D_{\text{unlearn}} \cup D_{\text{attack}}$ and then unlearn D_{unlearn}. After repeating this procedure three times, each target sample will have at least one observation instance for all intended distributions. Once all N iterations are completed, this structure guarantees that each target sample has at least $\frac{N}{3}$ complete set of observations to form the corresponding distributions $(\mathcal{Q}_u, \mathcal{Q}_R, \mathcal{Q}_h)$. The resulting distributions are then used to fit KDE distributions for the privacy leakage and efficacy evaluations of the target samples.

Algorithm 1 Membership Inference Attack in RULI

```
Input: trained model $f$, target sample $z \in D_{\text{target}}$, forget data $D_f \in D_{\text{attack}}$, data distribution $\mathcal{D}$, training algorithm $\mathcal{A}$, unlearning algorithm $\mathcal{U}$, inference function $\phi$ (loss, logit-scaled confidence), output observation $O$, number of shadow models $N$.
 1: Initialize: $O_{\text{in}}, O_{\text{out}}, O_{\text{unlearned}}, O_{\text{held-out}}$
 2: $O_{\text{remained}} \leftarrow \emptyset$
 ▷ Step 1: training & unlearning shadow models
 3: for $N$ iterations do
 4:     $D_{\text{attack}} \leftarrow \mathcal{D}$
 5:     $f_i \leftarrow \mathcal{A}(D_{\text{attack}} \cup \{z\})$
 6:     $f_R \leftarrow \mathcal{A}(D_{\text{attack}})$
 7:     $O_{\text{in}} \leftarrow O_{\text{in}} \cup \{\phi(f_i(z))\}$
 8:     $O_{\text{out}} \leftarrow O_{\text{out}} \cup \{\phi(f_R(z))\}$
 9:     $f_{\text{u}} \leftarrow \mathcal{U}(\mathcal{A}(D_{\text{attack}} \cup \{z\}), D_f \cup \{z\})$
10:     $O_{\text{unlearned}} \leftarrow O_{\text{unlearned}} \cup \{\phi(f_{\text{u}}(z))\}$
11:     $f_{\text{h}} \leftarrow \mathcal{U}(\mathcal{A}(D_{\text{attack}}), D_f)$
12:     $O_{\text{held-out}} \leftarrow O_{\text{held-out}} \cup \{\phi(f_{\text{h}}(z))\}$
13:     $f_{\text{r}} \leftarrow \mathcal{U}(\mathcal{A}(D_{\text{attack}} \cup \{z\}), D_f)$
14:     $O_{\text{remained}} \leftarrow O_{\text{remained}} \cup \{\phi(f_{\text{r}}(z))\}$
15: end for
 ▷ Step 2: estimating distributions
16: $\hat{f}_{\text{in}} \leftarrow \texttt{KDE}(O_{\text{in}})$
17: $\hat{f}_{\text{out}} \leftarrow \texttt{KDE}(O_{\text{out}})$
18: $\hat{f}_{\text{unlearned}} \leftarrow \texttt{KDE}(O_{\text{unlearned}})$
19: $\hat{f}_{\text{held-out}} \leftarrow \texttt{KDE}(O_{\text{held-out}})$
20: $\hat{f}_{\text{remained}} \leftarrow \texttt{KDE}(O_{\text{remained}})$
 ▷ Step 3: querying models
21: $f_U \leftarrow \mathcal{U}(f, z)$
22: $O_{f_U} \leftarrow \phi(f_U(z))$
23: $f_T \leftarrow$ Equation 2
24: $O_{f_T} \leftarrow \phi(f_U(z))$
 ▷ Step 4: deriving the privacy leakage & efficacy
25: $\Lambda \leftarrow \frac{p(O_{f_U} | \hat{f}_{\text{unlearned}})}{p(O_{f_U} | \hat{f}_{\text{held-out}})}$
26: $\Psi \leftarrow \frac{p(O_{f_T} | \hat{f}_{\text{unlearned}})}{p(O_{f_T} | \hat{f}_{\text{out}})}$
27: Return $\Lambda$, $\Psi$
```

RULI vs. U-LiRA [38]. One might naturally extend the MIAs to the unlearning setting by considering the unlearning step, comparing the distributions of retrained and unlearned models with a query from an unlearned model and formulating a likelihood-based ratio test such as:

$$\frac{p(\theta_{\mathcal{U}} \mid \mathcal{Q}_u(z))}{p(\theta_{\mathcal{U}} \mid \mathcal{Q}_R(z))}.$$

This is the core idea behind U-LiRA [38], the first per-sample framework for evaluating unlearning by comparing inferences from the unlearned model $(\mathcal{Q}_u)$ against those of a retrained model $(\mathcal{Q}_R)$. U-LiRA marks a notable step towards structured evaluation of unlearning effectiveness. Then, we summarize the major differences between RULI and U-LiRA as below.

Efficacy Modeling. The hypothesis test in U-LiRA (as discussed above) relies on a critical assumption: the inference behavior of the unlearned model, $\theta_{\mathcal{U}}(z)$, will closely match that of the retrained model, $\theta_R(z)$, for target samples $z \in D_{\text{target}} \setminus D_{\text{train}}$. Membership is then inferred by comparing the unlearned model's output at z to the distribution $\mathcal{Q}_R(z)$.

However, this assumption may not hold. Our empirical analysis (see Figure 9) reveals a distributional mismatch: the unlearned and retrained models diverge significantly where z is out. This divergence is not merely a subtle shift—it directly impacts MIA performance and may lead to impacting privacy leakage and more noticeably efficacy measurement. Also, this mismatch reflects a limitation of Game 2 and our motivation for introducing Game 3 and consequently the Test model for efficacy measurement in RULI.

Targeted Attack for Unlearning Evaluation. First, in contrast to U-LiRA, RULI incorporates an additional shadow distribution $\mathcal{Q}_h$, which represents held-out samples, i.e., samples that are neither trained nor unlearned. This enables a dual-objective inference framework where privacy leakage Λ is evaluated against $\mathcal{Q}_h$, and efficacy Ψ is evaluated against $\mathcal{Q}_R$ (not in U-LiRA). Second, this requires a revised shadow model design, and RULI remains computationally efficient by sharing models across roles. Third, RULI involves a new target sample selection strategy on vulnerable samples (canaries) while U-LiRA focuses on random samples. We demonstrate that applying our target selection strategy to U-LiRA improves its MIA performance, outperforming previously reported accuracy even for strong unlearning baselines. Finally, as a targeted attack, RULI requires significantly fewer shadow models, making it more efficient than U-LiRA. We provide empirical comparisons for RULI and U-LiRA in Section 6.4.

5 Rectified Unlearning Evaluation

In this section, we integrate the proposed MIA in RULI into a unified framework for unlearning evaluation. Section 5.1 illustrates how these components can interconnect to provide a holistic view of privacy risks and efficacy. With the refined likelihood ratios, we then perform targeted attacks on canaries using the selection strategy from Section 5.2.

5.1 Major Steps for Rectified Evaluation

Shadow Model Training. Using the MIA in RULI, we train N shadow models with a fixed target unlearning set. For the shadow-trained models, we obtain the *In* and *Out* distributions. For the shadow-unlearned models, each sample $x \in D_{\text{target}}$ is evenly distributed across three inference types: *remained*, *unlearned*, and *held-out*, with each type represented by $\frac{N}{3}$ models. This corresponds to Steps 1 and 2 in Algorithm 1.

Target Model Training and Unlearning. We begin by partitioning the unlearning target set D_{target} into three equal subsets: one-third is designated to be excluded from training, while the remaining two-thirds are used to train the target model θ_I along with disjoint attack data to ensure generalization. After training, we perform unlearning by removing half of the D_{target} training subset, resulting in the unlearned model $\theta_{\mathcal{U}}$. This setup ensures a balanced design for evaluating both validation accuracy and membership inference (MI) using the two likelihood tests defined in RULI: Λ (privacy leakage) and Ψ (unlearning efficacy), as detailed in Steps 3 and 4 in Algorithm 1. Since these evaluations require equal-sized subsets for fair comparison, we may adjust the partitioning ratio of D_{target} accordingly if needed.

Note that RULI can provide a comprehensive set of likelihood tests in Table 1 with a single run. It enables measurement metrics other than privacy leakage and efficacy on forget data. For example, we might evaluate how unlearning impacts remain data privacy and how memorization levels would be changed. Ideally, the memorization on remain samples should remain similar to the original model (no unintended privacy leakage and no unintended unlearning).

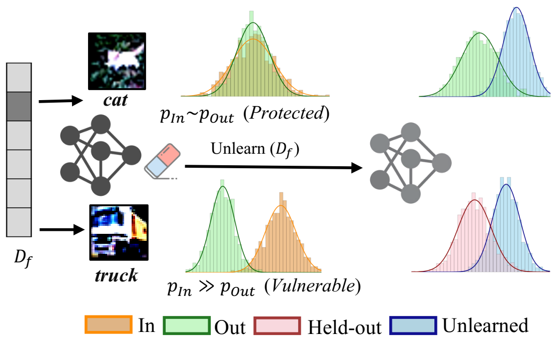
Table 1: RULI enables comprehensive MIA tests on unlearned and trained models.

Target Evaluation	Likelihood Ratio Test
Unlearning efficacy	$\frac{p(\theta_{\mathcal{T}} \mid Q_u(z))}{p(\theta_{\mathcal{T}} \mid Q_R(z))}, \quad \forall(z) \in D_{\text{target}}$
Privacy leakage	$\frac{p(\theta_{\mathcal{U}} \mid Q_u(z))}{p(\theta_{\mathcal{U}} \mid Q_h(z))}, \quad \forall(z) \in D_{\text{target}}$
Trained model privacy leakage	$\frac{p(\theta_I \mid Q_i(z))}{p(\theta_I \mid Q_R(z))}, \quad \forall(z) \in D_{\text{target}}$
Privacy leakage on remained samples	$\frac{p(\theta_{\mathcal{U}} \mid Q_r(z))}{p(\theta_{\mathcal{U}} \mid Q_h(z))}, \quad \forall(z) \in D_{\text{target}}$

5.2 Target Samples Selection

To bridge the proposed attack with the evaluation, we now address a critical challenge: How to select target samples that represent corner cases in unlearning? Most existing unlearning methods rely on random subsets of the training data, focusing on average-case evaluations. However, not all samples are memorized equally, and most are generally well-protected in typical datasets [71]. This inspires us to investigate unlearning in a more challenging scenario: How does unlearning perform on highly memorized (vulnerable) samples?

We further examine the impact of unlearning vulnerable samples alongside non-vulnerable (i.e., safe or protected) ones, inspired by the canary injection technique from privacy auditing [14, 52, 66]. Our results show that, when carefully tuned, stronger unlearning methods can provide greater protection than expected—especially *when only vulnerable samples are unlearned* (see Table 2 and Table 5). We found that a more practical setting of injecting vulnerable samples as canaries yields higher privacy leakage and lower efficacy (higher MIA success). We suspect this behavior is influenced by the relative impact of sample vulnerability, and when samples with different memorization levels are unlearned in mini-batches, the averaged gradient cannot update the model to sufficiently unlearn the vulnerable samples. An example of such an effect in Figure 3 where unlearning vulnerable samples together with protected samples is challenging. We leave a deeper investigation of this observation to future work. For now, we consider this scenario the most challenging and practical setting for unlearning. To support this, we have conducted a comprehensive study on alternative target sample selection strategies in Section 6.2.1.



cat
$p_{In} \sim p_{Out}$ (Protected)
Unlearn (D_f)
D_f
truck
$p_{In} \gg p_{Out}$ (Vulnerable)
In
Out
Held-out
Unlearned

Figure 3: Two samples under inexact unlearning; A protected sample's out and unlearned distributions are distinguishable (efficacy); vulnerable sample's unlearned and held-out distributions are distinguishable (privacy leakage).

Canary Injection. By specifying the target set in RULI, we can focus on measuring privacy leakage (rather than efficacy), drawing inspiration from the well-known canary injection technique [14, 66] used in differential privacy auditing [52]. This approach offers an efficient strategy for evaluating inexact unlearning algorithms on their ability to remove vulnerable samples. The strategy is simple: inject canaries into the forget set and query only these samples after unlearning.

Specifically, we inject vulnerable samples into a randomly selected forget set. This random forget set may or may not overlap with a predefined set of *safe* samples, allowing us to simulate realistic unlearning scenarios (a portion of forget data can be vulnerable). Then, we only focus on the attacker's advantage solely on vulnerable samples. However, one might try to achieve tighter bounds on the privacy leakage of unlearning by a more careful choice of canaries or adversarial canary injection. In this paper, we focus on providing more insights for unlearning evaluation rather than trying to achieve very tight bounds on the privacy leakage of unlearned models.

6 Experiments

6.1 Experimental Setup

Consistent with existing research on unlearning [34, 38, 44, 46, 70] and machine learning privacy benchmarks [12, 38, 48], we evaluated the unlearning methods on typical CIFAR-10 and CIFAR-100 datasets with ResNet models. To demonstrate the generalizability of RULI, we evaluate unlearning on two challenging settings: TinyImageNet fine-tuned on a vision transformer (ViT), and WikiText-103 fine-tuned for text generation on two language models. We will provide the experiment setup in the following as we proceed. Details on the training hyperparameters are deferred to Appendix A.1.

6.1.1 Unlearning Benchmarks

We adopted RULI and other MIAs to attack and evaluate the following SOTA unlearning methods.

ℓ_1 **Sparse** [46] based unlearning approach employs a *pruning and fine-tuning* strategy. Initially, it removes weights with the smallest absolute values, based on the assumption that these weights contribute the least to the model's performance. This pruning step not only facilitates unlearning but also results in a sparser, potentially smaller neural network. After pruning, the model is fine-tuned on D_r to recover its performance.

Scrub [44] employs a distillation-based approach for unlearning. It guides the model to diverge from the original predictions for D_f. Simultaneously, it ensures that the model's predictions on D_r remain consistent with its original behavior. By leveraging distillation, Scrub achieves a balance between unlearning the forget data and preserving the accuracy and functionality of the remain data.

GA/GA+ [34, 44, 70] is an enhanced version of gradient ascent tailored for unlearning. It operates in two stages: first, it maximizes the loss on D_f, then, it fine-tunes the model to minimize the loss on the D_r, ensuring that the model maintains its predictive accuracy on remain data. We specifically add fine-tuning (refining) steps to make gradient ascent comparable with other benchmarks.

NegGrad+ [38, 44] is designed to optimize two goals: raising the loss on the forget set while reducing the loss on the remain data, controlled by a balancing parameter.

There are growing baselines not included in this paper, as the methods presented above consistently achieved superior performance, in line with the findings of [38, 46]. However, any inexact unlearning algorithm can be measured by RULI.

6.1.2 Target Samples for Unlearning

We collected target samples for unlearning by conducting classic Membership Inference Attacks (MIAs), specifically LiRA [12]. For a given sample, $p(In)$ is the estimated likelihood probability that it was in the training set, while $p(Out)$ is the estimated probability that it was not. Samples with likelihoods exceeding a certain threshold at TPR@lowFPR are marked as highly memorized (i.e., vulnerable). Conversely, when $p(In) \approx p(Out)$, the samples are considered less memorized, making them more resistant to MIAs. Specifically, for CIFAR-10 and CIFAR-100, we consider the standard setting from [12], where a model is trained on a dataset containing half of the samples from training set (i.e., 25,000 training samples). The remain data serves as the attack dataset in Section 6.1.3. We then identified the vulnerable highly memorized samples (easier to attack) and protected samples as the ones with lower memorization (harder to attack). We refer readers to more details in A.3

Target Samples for Shadow Model Training and Unlearning. We select different setups of target samples and name these settings in the parentheses: 1) 600 random samples ("Random"), 2) 600 vulnerable samples ("Vulnerable"), 3) 600 protected samples ("Protected"), 4) 600 protected samples + 600 vulnerable samples ("Vulnerable + Protected"), and 5) 600 random samples of one class of dataset ("Class").[2]

6.1.3 MIA Settings

After performing model unlearning, we evaluate the unlearned models using MIAs.

Shadow Model Training. To facilitate MIAs, we construct a training set that includes unlearned examples from the various unlearning settings, along with randomly sampled data from the attack dataset, which is non-overlapping with the unlearning set.

For MIA, we trained 90 shadow models, each for 50 epochs. The shadow models achieve an accuracy of at least 88%. Additionally, we tune hyperparameters to ensure that the unlearning baselines match the *Retrain* gold standard in accuracy across the remain, forget, and test datasets [74]. This involves conducting a comprehensive grid search similar to [38], but with a set of parameters specifically adapted to each target data setup (see Table 9 in Appendix A.2 for details). Our implementation leverages the open-sourced code bases from these benchmarks.

Average-case Attack. To implement an average-case attack as a baseline, we aggregate the logit-scaled confidences across all shadow models following similar population attack adaptations [35, 38, 62]. Specifically, for each setting (e.g., when the target data consists of vulnerable samples), we combine the outputs from all shadow models to form aggregated populations. A linear binary classifier is then fitted to these aggregated distributions, learning to differentiate between member and non-member populations across the entire set.

[2]Hayes et al. [38] considers this setting as a challenging setting for CIFAR-10; We have excluded class-wise unlearning as removing an entire class from the dataset violates typical MI assumptions (blind attacks would outperform any targeted attack in such a setting).

6.2 Privacy: (Stand-alone) Unlearned Model

6.2.1 Privacy Leakage across Different Target Sets

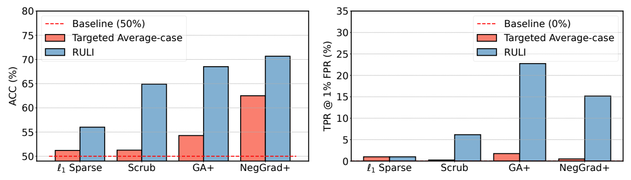
We evaluate RULI under the standard setting that we have shadow models for all target sample setups. This study helps illustrate which unlearning settings are more prone to our privacy attacks. As shown in Table 2, we report Attack Accuracy, Attack AUC, and TPR@1% FPR as privacy leakage metrics for different unlearning methods, considering different combinations of forget and target data. Additionally, we present the accuracy gap between the unlearned model ($\theta_{\mathcal{U}}$) and the retrained model ($\theta_{\mathcal{R}}$) to provide a general performance evaluation of the unlearning methods.

Table 2: Attacking or evaluating unlearning methods on the CIFAR-10 dataset (whether by selecting safe samples, class-specific random samples, or exclusively vulnerable samples as forget sets, does not effectively assess privacy risk–**Pitfall II**). A better approach is to use target samples as canaries. Inferences are performed in a single training and unlearning run, selecting the best unlearning instance for evaluation.

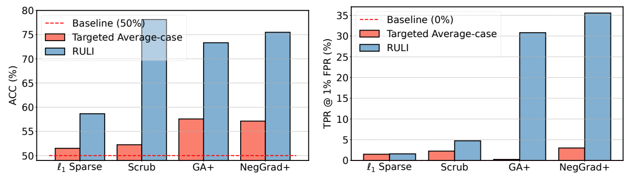
Unlearning Methods	Forget Data	Target Data	Attack AUC	Attack ACC	TPR@ 1% FPR	Δ ACC(D_f)
ℓ_1 Sparse	+ Random	Random	45%	48%	1%	2.5%
	+ Class	Class	54.0%	**53.75%**	0.0%	7.5%
	+ Vulnerable	Vulnerable	55.97%	50.50%	1%	3%
	+ (Vulnerable + Protected)	Protected	48.96%	48.46%	1.06%	4%
	+ (Vulnerable + Protected)	Vulnerable	**56.85%**	53.37%	**1.9%**	
Scrub	+ Random	Random	52%	51.75%	3%	0.5%
	+ Class	Class	61%	57%	1%	0.3%
	+ Vulnerable	Vulnerable	57.0%	54.5%	5%	3.0%
	+ (Vulnerable + Protected)	Protected	57.75%	49.98%	0.0%	7.5%
	+ (Vulnerable + Protected)	Vulnerable	**73.53%**	**68.27%**	**8.53%**	
GA+	+ Random	Random	56%	54.75%	7%	10.5%
	+ Class	Class	61%	55.25%	3%	7%
	+ Vulnerable	Vulnerable	60.75%	65%	6%	10%
	+ (Vulnerable + Protected)	Protected	52.13%	54.17%	0.0%	7.75%
	+ (Vulnerable + Protected)	Vulnerable	**76.17%**	**68.5%**	**22.75%**	
NegGrad+	+ Random	Random	52%	52.75%	4%	2%
	+ Class	Class	69%	58.50%	4%	17%
	+ Vulnerable	Vulnerable	76%	70.25%	21%	43.5%
	+ (Vulnerable + Protected)	Protected	52.5%	51.56%	1.59%	15.5%
	+ (Vulnerable + Protected)	Vulnerable	**77.54%**	**72.36%**	**18.48%**	

The results show that privacy leakage is more pronounced when the forget set includes a mix of vulnerable and protected samples, compared to using random samples or random samples from "Class". For example, in GA+, when the forget data and target data consist of random samples or random samples from "Class", the attack AUC is 56% and 61%, respectively, while TPR@1% FPR is 7% and 3%. Similarly, the attack accuracy is 54.75% and 55.25%. However, when the forget data consists of both Vulnerable and Protected samples and the target data is Vulnerable, the attack AUC, attack accuracy, and TPR@1% FPR increase significantly to 76.17%, 68.5%, and 22.75%, respectively. A similar trend is observed for Scrub and NegGrad+, indicating that RULI is highly effective at inducing privacy leakage across most unlearning methods. We attribute this behavior to the batch-averaging effect inherent in methods like NegGrad+, GA+, and Scrub, which optimize an objective loss on D_f (distillation for Scrub and cross-entropy for NegGrad+ and GA+). As expected, ℓ_1 Sparse demonstrates the highest resilience to RULI, likely due to its reduced memorization capacity, which makes it less vulnerable to both RULI and traditional MIAs [40] on model training. However, this resilience comes at a cost, as sparse models also exhibit reduced memorization of remaining vulnerable samples, a tradeoff discussed further in Section 6.3. Overall, we conclude that using a combination of Vulnerable + Protected samples as forget data and Vulnerable samples as target data represents the most effective setting for evaluating privacy leakage.

Additionally, we observe that GA+ achieves a similar accuracy gap (7.75% vs 7.5%) compared to Scrub but exhibits a significantly higher TPR@1% FPR, with a margin of 14.22%. This highlights the inadequacy of relying on average accuracy to assess unlearning privacy leakage.



(a) CIFAR-10 Results



(b) CIFAR-100 Results

Figure 4: Attack accuracy and TPR@1%FPR targeting vulnerable samples as canary with RULI vs targeted average-case attack. Average-case attacks are underestimating the privacy risk (TPR@1%FPR) even under target vulnerable samples. RULI achieves higher attack success on all methods.

6.2.2 Vulnerable Samples as Canaries

We examine vulnerable samples analogous to canaries [3, 66] in a similar setting. We inject 600 vulnerable samples (100 would be trained and remain in the model to prevent possible *Onion effect* [13]) along with 600 random samples, except for the vulnerable set. Consequently, we will have *500 canaries to target (250 unlearned and 250 held-out)*.

Targeted Average-case Attack. As a baseline vs RULI, we adapt average-case attacks to evaluate its performance. Specifically, we derive two populations from all distributions generated by shadow models for the vulnerable samples. Using these populations, we train a regression model to classify the samples based on their population membership. The model can distinguish between samples that were unlearned and those that were entirely held out from both the training and unlearning processes. This approach shows RULI's effectiveness in capturing nuanced privacy leakage.

Results. Figure 4 presents the attack accuracy and TPR@1% FPR for CIFAR-10 and CIFAR-100 datasets, where RULI outperforms average-case attacks significantly. For CIFAR-10, across the unlearning benchmarks, the attack accuracy and TPR@1% FPR achieved by RULI are, on average, 18.63% and 11.88 times higher, respectively, compared to the average-case attacks. Similarly, for CIFAR-100, RULI surpasses average-case attacks by 30.72% in attack accuracy and 9.38 times in TPR@1% FPR. Thus, the average-case attacks significantly underestimate privacy risks, even when targeting vulnerable samples, underscoring the severity of **Pitfall I**.

We acknowledge that our strategy does not offer the tightest privacy leakage estimation. For example, leveraging only the top vulnerable samples and getting the membership inference over hundreds of runs on those samples, [3, 66] can potentially provide higher TPR@low FPR results. We leave this exploration to future works, positioning our findings as a step towards systematically attacking unlearning algorithms.

6.3 Efficacy: Unlearning vs. Retraining

We evaluated RULI for efficacy attack to determine whether a target sample was unlearned or retrained, as described in Game 3. To achieve this, we utilized Algorithm 1 to compute the likelihood test ratio in accordance with Equation 3. This experiment requires Test model to construct a hypothetical test model, as defined in Equation 2 and illustrated in Figure 2. The goal of this experiment is to provide a more fine-grained evaluation for comparing the effectiveness of inexact unlearning methods against the retraining global standard.

We consider four attack settings, where the target data consist of "Random" Samples, "Vulnerable" Samples, "Vulnerable + Protected" Samples, and randomly selected samples from "Class", respectively. Figure 5a-5d, demonstrate that by querying the test model $\theta_{\mathcal{T}}$, the attacker achieves a high attack success rate, effectively distinguishing many samples under TPR@FPR. From the figures, we observe that in our newly introduced attack settings (Vulnerable and Vulnerable + Protected), TPR@FPR is larger (TPR@1%FPR$\geq$10%) across all unlearning methods. Notably, the Vulnerable + Protected setting dominates on the ℓ_1 Sparse and Scrub. The Vulnerable setting achieves better results on GA+ and NegGrad+. These findings indicate that RULI successfully avoids **Pitfall III**, effectively distinguishing retrained models from unlearned models. While these results do not show the privacy leakage, they establish a promising foundation also for future works. One could leverage this method to audit the upper bounds of certified unlearning approaches [18, 36, 81].

Some Insights. Efficacy attack can be used as the definitive evaluation standard, as it provides a clear basis for comparing inexact unlearning with retraining. An interesting finding that is not observable in average-case accuracy evaluation is the unintended loss of memorization of remained vulnerable samples caused by inexact unlearning algorithms. Ideally, only the information related to the forget set should be removed. However, we observed that the model's accuracy on the remaining vulnerable dataset was also unintentionally reduced.

Table 3: Inexact unlearning methods cannot accurately preserve the remained vulnerable samples (shown as Vul set).

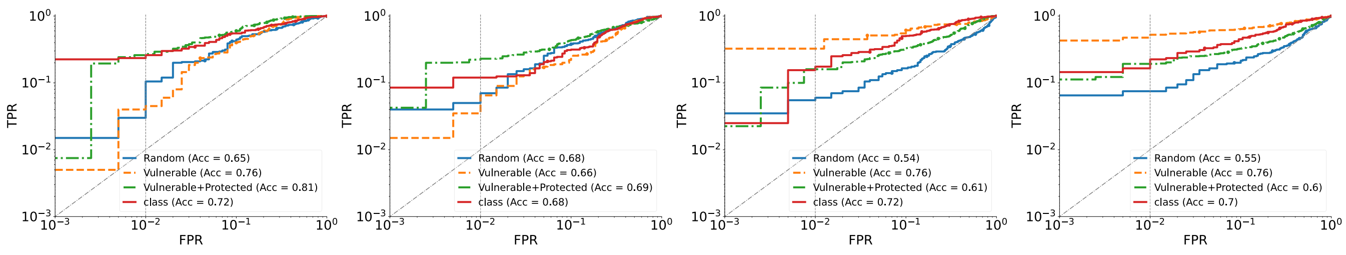
Unlearning Methods	CIFAR-10				CIFAR-100			
	$Acc(D_f)$	$Acc(D_r)$	$Acc(D_{\text{test}})$	$Acc(D_r)$ on Vul set	$Acc(D_f)$	$Acc(D_r)$	$Acc(D_{\text{test}})$	$Acc(D_r)$ on Vul set
ℓ_1 Sparse	57.5%	94.84%	88.36%	**45.10%**	51%	87.48%	66.34%	**50.54%**
Scrub	65.0%	97.58%	89.63%	70.10%	61.5%	96%	64.86%	83.70%
GA+	66.60%	94.42%	84.24%	62.50%	67%	99.11%	63.13%	100%
NegGrad+	81.0%	96.93%	85.74%	80.96%	65.0%	91.89%	58.41%	86.96%
Retrain	59.50%	99.79%	90.71%	96.19%	35.75%	99.93%	67.39%	99%

As shown in Table 3, $ACC(D_f)$, $ACC(D_r)$, $ACC(D_{\text{test}})$, and $ACC(D_r)$ on Val set denote accuracy on forget data, remain data, test data, and remaining vulnerable data, respectively. Generally, after unlearning, the model performs well on forget data, remain data, and test data, as the $ACC(D_f)$, $ACC(D_r)$, $ACC(D_{\text{test}})$ of unlearned models are comparable to those of retrained models for both datasets. However, unlearned models show a significant accuracy drop on remaining vulnerable data compared to the retrained model. For CIFAR-10, $ACC(D_r)$ on the vulnerable set is 96.19% for the retrained model but only 64.67% on average for unlearned models. Similarly, for CIFAR-100, $ACC(D_r)$ on Val set is 99% for the retrained model versus 80.3% on average for unlearned models.

Among the benchmarks, ℓ_1 Sparse exhibited the most pronounced loss of memorization on the remain data. We hypothesize that this is due to the strong global sparsity enforced by the method. While the sparsity was maintained at approximately 60% to optimize the accuracy gap, it likely contributed to the observed loss in memorization. Notice that, to validate our findings, we reduce model sparsity to 7%, observing only a 1–2% change in accuracy on the remaining vulnerable set.

6.4 Comparison with U-LiRA

Differences in Target Design and Testing Strategy. RULI differs from U-LiRA along two key designs: target sample selection and model querying during hypothesis testing. U-LiRA selects random samples from a single class (the setting of "Class"), while we construct the target set by injecting vulnerable samples into a set of known protected samples, though we only evaluate on the vulnerable ones. Additionally, while both U-LiRA and RULI query the unlearned model $\theta_{\mathcal{U}}$ for privacy leakage, they differ in how reference distributions are constructed. U-LiRA compares outputs between unlearned and retrained models ($\mathcal{Q}_u$ vs $\mathcal{Q}_R$), while RULI uses $\mathcal{Q}_h$, a held-out distribution $\mathcal{Q}_R$. For efficacy, RULI evaluates queries using the test model $\theta_{\mathcal{T}}(z)$, which selects between a



(a) ℓ_1 Sparse (b) Scrub (c) GA+ (d) NegGrad+

Figure 5: Comparing with retraining using RULI. All unlearning benchmarks are highly distinguishable from retraining.

retrained model $\theta_{\mathcal{R}}$ or an unlearned model $\theta_{\mathcal{U}}$ depending on whether the sample was excluded or unlearned.

Table 4: Ablation study comparing U-LiRA and RULI under different target selection strategies and MIAs for Scrub and ℓ_1 Sparse on CIFAR-10 dataset. *For "Vulnerable + Protected", privacy leakage is computed only on vulnerable samples.* Results are averaged over 10 independent runs.

Setting	Target Samples	MIA Method	Privacy Leakage		MIA for Efficacy	
			ACC ↑	TPR@1 %FPR ↑	ACC ↑	TPR@1 %FPR ↑
ℓ_1 Sparse						
U-LiRA Baseline	one "Class" only	U-LIRA	50.2%	0.5%	-	-
Alt. MIA Only	Vulnerable + Protected	U-LiRA	55.7%	1.6%	-	-
Alt. Target Only	one "Class" only	RULI	51.73%	1.37%	65.6%	18.0%
RULI (Full)	Vulnerable + Protected	RULI	54.4%	1.4%	81.1%	23.3%
Scrub						
U-LiRA Baseline	one "Class" only	U-LIRA	53.75%	1.13%	-	-
Alt. MIA Only	Vulnerable + Protected	U-LiRA	62.8%	8.6%	-	-
Alt. Target Only	one "Class" only	RULI	53.32%	1.7%	65.1%	7.0%
RULI (Full)	Vulnerable + Protected	RULI	65.5%	11.8%	67.4%	8.8%

Results.[3] Table 4 evaluates the impact of two key components in our evaluation pipeline: (1) the target sample selection strategy, and (2) the MIA. We compare four configurations using both ℓ_1 Sparse and Scrub unlearning (*which achieve the top unlearning performance from prior results*). U-LiRA's original setting selects random samples from "Class" as the target set and uses their own likelihood-ratio test. In contrast, our setting structures the target by injecting vulnerable samples into a set of protected samples, and uses RULI for attack.

When we replace only the target selection while keeping U-LiRA's test, privacy leakage improves significantly. Under Scrub, TPR@1%FPR increases from 1.13% to 8.6%. This shows that our target selection alone improves the evaluation, even benefiting U-LiRA. Conversely, when we change only the attack to RULI while keeping the "Class" target, the gains are marginal, suggesting that these targets limit the attack performance. Finally, in our full setting (Vulnerable + Protected target + RULI), we observe the strongest results, e.g., Scrub achieves 11.8% TPR@1%FPR for privacy leakage and 8.8% for efficacy, while ℓ_1 Sparse reaches 1.4% and 23.3%, respectively. Since U-LiRA assesses efficacy through the lens of privacy leakage, we do not report MIA results for its efficacy.

Efficiency. We acknowledge that training shadow models introduces additional computational costs, and this applies to all per-sample standard MIA evaluations. For reference, RULI incurs 1.2× runtime of LiRA [12] (standard per-sample MIA for machine learning training) per shadow model under unlearning CIFAR-10 using Scrub. In practice, U-LiRA is more computationally expensive. For example, in a setting with 1,200 target samples (e.g., our Vulnerable + Protected setting), for preparing 30 unlearned shadow models per sample, U-LiRA incurs approximately 270× higher cost than single shadow model training and requires around 40× unlearning models for each trained model. In contrast, RULI completes shadow models with 36× more than single shadow model training.

[3]We re-implemented U-LiRA (not official) based on the algorithm in [38].

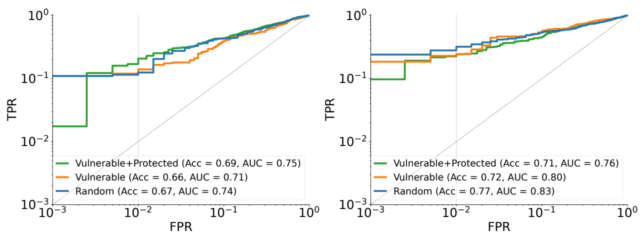
6.5 Generalizability of RULI

We next evaluate the generalizability of RULI by applying it to model fine-tuning on larger-scale and complex datasets (TinyImageNet and WikiText-103) and unlearning in vision transformer (ViT) and language models.

6.5.1 Vision Transformer (ViT) on TinyImageNet

For the TinyImageNet dataset, we employ a pre-trained ViT (Swin-small [47] Transformer) to fine-tune and unlearn. Training is performed to reach the test accuracy of at least 84% by fine-tuning the model only with 2 epochs. Shadow model training and canary injection settings are similar to those on preparing 90 shadow models in Sections 6.1.3 and 6.2.2. For unlearning evaluation, we avoid from-scratch training or fine-tuning architectures like ResNets on this dataset, due to low test accuracy and severe overfitting, making them unsuitable for our study. We also use the top benchmarks from prior results (ℓ_1 Sparse and Scrub) to unlearn image samples from the ViT model for this group of experiments.

Results. Efficacy results indicate a consistent gap in per-sample behavior between the retrained model and unlearned models across all three target data configurations, achieving an attack accuracy of at least 66% on ℓ_1 Sparse and 71% on Scrub. Efficacy tests can achieve at least 10% TPR@1%FPR in the "Vulnerable + Protected" setting. The privacy leakage results show a different trend. We conduct a targeted popu-



(a) ℓ_1 Sparse (b) Scrub

Figure 6: MIA for unlearning efficacy on different choices of target samples, with best unlearning baselines evaluated when fine-tuning and unlearning a Swin-small ViT model.

Table 5: Privacy leakage on TinyImageNet. We evaluate three target selection strategies—*Canary injection*, *Vulnerable-only*, and *Random samples*—using a target set of 500 samples (250 unlearned and 250 are held-out). All experiments are performed on a Swin-small ViT model.

Target data	Targeted average-case attack (Population attack)				RULI			
	AUC	ACC	TPR@ 1%FPR	TPR@ 5%FPR	AUC	ACC	TPR@ 1% FPR	TPR@ 5%FPR
ℓ_1 Sparse								
Vulnerable only	54.4%	55.1%	2.3%	5.2%	59.6%	56.0%	2.4%	12.4%
Vulnerable as canaries	55.3%	54.7%	0.8%	5.6%	**62.6%**	**57.0%**	**6.3%**	**16.6%**
Random	53.2%	52.8%	0.0%	2.4%	56%	54.4%	0.8%	6.4%
Scrub								
Vulnerable only	52.5%	52.4%	2.0%	5.4%	65.3%	61.5%	11.7%	23.9%
Vulnerable as canaries	56.0%	56.2%	1.0%	6.3%	**69.5%**	**63.6%**	**10.9%**	**27.1%**
Random	49.6%	49.8%	1.0%	2.8%	59.7%	57.0%	6.0%	14.0%

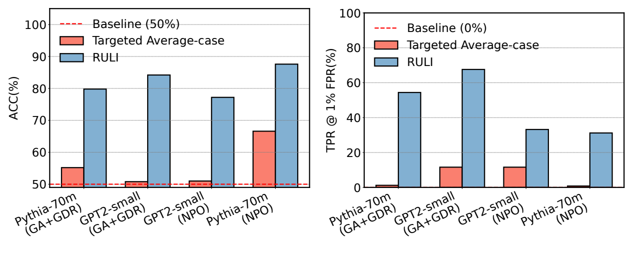
lation attack as the baseline and compare it with RULI on a different set of target data. Results confirm the prior observations: unlearning vulnerable samples, when injected as canaries, leads to increased privacy leakage. Exclusively unlearning vulnerable samples results in a higher attack success rate compared to random samples, but is not generally more effective than when they are injected as canaries.

6.5.2 Text Generation on WikiText-103

We conduct our experiments on the WikiText-103 dataset, fine-tuning Pythia-70m [8] and GPT2-small [57] (see details in Appendix A). The unlearning task focuses on removing *n-gram sequences* that the fine-tuned models have memorized within the training. To validate RULI's generalizability, we apply unlearning methods designed for language models [43, 45, 61, 82]. The attacker aims to infer the membership of the specific n-gram sequence by querying the unlearned model.

Unlearning Benchmarks. Previous inexact unlearning methods in Section 6.1.1, are not originally proposed for language model unlearning. Instead, we adapt two gradient-based unlearning methods [61]: GA+GDR and NPO [82] to unlearn sequences from language models. GA+GDR applies Gradient Ascent on the forget data to increase its loss and reduce its influence, while using Gradient Descent on the remain data (GDR) to preserve generalization. NPO introduces a directional forgetting loss that compares the current model's logits to those of a frozen trained model (*reference model*), directing the updated model to assign lower confidence to the forget data. In both methods, the forgetting and retention objectives are applied jointly to ensure effective removal without degrading overall performance. Moreover, we add SFT steps on remaining data to recover any loss due to unlearning.

Attack Setup. For shadow model training, we use a dataset of 15,000 records (containing over 1 million tokens) and consider 1,000 7-gram sequences [24] (with prefixes drawn from test data records) as the target data for training shadow models. We fine-tune both Pythia-70m and GPT2-small with 5 supervised fine-tuning (SFT) [10] epochs and 2 epochs of prefix language modeling (PLM) [58]. Similarly, for unlearning, we use PLM to optimize the model to unlearn the target 7-grams. We also execute 2 more SFT epochs on the remaining data to improve unlearning and generalization (see more details in Appendix A). We have trained 60 shadow models and collected the target 7-grams samples' loss [55, 78] for membership signaling. Note that canary injection was not applied, as it is not yet commonly adopted in language models; nonetheless, RULI still demonstrates strong performance in evaluating text unlearning, as detailed below.

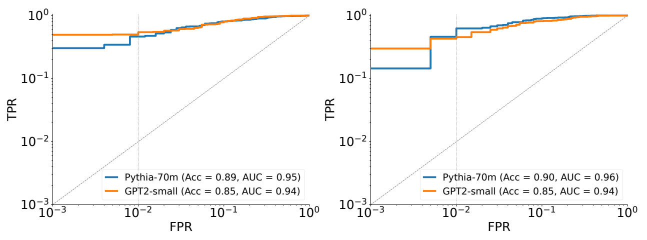


(a) Attack accuracy results (b) TPR@1%FPR results

Figure 7: Privacy leakage results on WikiText-103 using a target set of 500 samples (*250 out and 250 unlearned*) on WikiText-103 with the unlearning 7-gram sequences task.

Results. Figure 7 shows that RULI achieves higher accuracy and TPR@1%FPR than the targeted average-case baseline across all evaluated models and unlearning methods. The Pythia-70m model generally yields more privacy leakage than GPT2-small, especially under the GA+GDR unlearning method. Figure 8 shows that both GA+GDR and NPO exhibit substantial differences from the retrained model, with Pythia-70m reaching up to 90% MIA accuracy. This suggests that the unlearned model still retains distinguishable characteristics compared to the retrained model.

Scaling to LLMs. While we demonstrate the effectiveness of RULI in multiple domains, including TinyImageNet and WikiText-103, there remain several challenges in applying it



(a) Attack accuracy results (b) TPR@1%FPR results

Figure 8: Unlearning efficacy on 400 target samples (*200 Out and 200 Unlearned*), evaluated using NPO and GA+GDR baselines with fine-tuning-based unlearning on GPT2-small and Pythia-70m models.

to production-scale language models with broad knowledge coverage. First, performing per-sample MIAs on LLMs is computationally intensive and may be impractical at scale. Second, privacy and unlearning on text data and LLMs have not yet been well standardized [63], and current membership-based auditing methods may not fully capture privacy leakage in LLMs [51]. Thus, while RULI is applicable to relatively smaller-scale settings for unlearning text in language models [55], privacy and efficacy evaluation for LLM unlearning needs further advancements, along with the future development of unlearning guarantees and privacy attacks on LLMs.

7 Related Works

Machine Unlearning. Numerous studies have explored machine unlearning, broadly categorized into data-oriented and model-oriented approaches [77].

First, data-oriented methods focus on modifying training data. The SISA framework by Bourtoule et al. [9] partitions data into subsets, enabling selective retraining. Tarun et al. [69] proposed adding error-maximizing noise to obscure forgotten samples. Second, model-oriented methods modify the model directly. Golatkar et al. [6, 34] used Fisher information [50] for selective forgetting, and later developed quadratic unlearning methods [33], though at some cost to accuracy. Chundawat et al. [21] introduced a student-teacher approach, and their zero-shot Gated Knowledge Transfer (GRT) method [22] uses pseudo data with a band-pass filter to isolate retained knowledge. Similar post-hoc model adjustment techniques have been developed for generative models. All inexact unlearning baselines in the evaluation fall in this category.

Furthermore, a parallel line of work investigates certified unlearning, which provides formal guarantees that unlearning approximates retraining without the removed data [18, 36]. These methods, inspired by differential privacy, define (ε, δ)-bounded differences in model outputs, often focusing on loss or gradient divergence. While theoretically sound, certified unlearning relies on conservative assumptions and tight constraints, making it less scalable to large models. Thus, we focus on evaluating practical, inexact unlearning methods, where empirical privacy risks are more pronounced.

Other Threats of Machine Unlearning. Although machine unlearning is designed to enhance data privacy, recent work has uncovered significant privacy risks. Most existing attacks operate under a strict threat model. Chen et al. [17, 31] showed that differences between models before and after unlearning can unintentionally leak information, enabling average-case MIA-based attacks to succeed. Under the same assumptions, [39] demonstrated model inversion attacks to further expose unlearning vulnerabilities. Attribute inference attacks [5, 30], which infer sensitive features from observed data, have also been adapted to unlearning. Stock et al. [67] investigated such attacks in the context of feature-level unlearning in white-box settings, evaluating whether specific attributes were present in the original dataset. Model inversion attacks [29] pose additional risks by reconstructing input data from model outputs. These attacks are particularly effective against models subjected to class-level unlearning [6]. Graves et al. [35] advanced this line of work by adapting model inversion techniques to assess the robustness of class-level unlearning strategies. Other lines of attack involve injecting poisoned data or issuing adversarial unlearning requests to disrupt the unlearning process [41]. These attacks aim to corrupt the unlearning mechanism and represent valuable, emerging directions; in contrast, our work focuses on evaluating whether unlearning has occurred under an assumed correct process.

8 Conclusion

In this work, we have proposed RULI, a novel framework that fills critical gaps in evaluating privacy and efficacy of inexact unlearning. RULI employs a dual-objective attack to provide fine-grained insights, outperforming existing methods in distinguishing unlearned models from retrained models and exposing privacy risks in SOTA benchmarks. We have validated its effectiveness on image (e.g., CIFAR-10/100, Tiny-ImageNet) and text (e.g., WikiText-103) datasets using ViT and language models. Our findings reveal persistent vulnerabilities, particularly for sensitive samples, and highlight the limitations of existing unlearnings. RULI sets a new evaluation standard for robust, scalable unlearning.

Acknowledgments

We sincerely thank the anonymous shepherd and all the reviewers for their constructive comments. This work is partially supported by the National Science Foundation (NSF) under Grants No. CNS-2432533, CNS-2308730, CNS-2302689, CNS-2319277, CNS-2241713, CNS-2331302, CNS-2339686, CNS-2029038, and CNS-2135988, as well as by a Cisco Research Award and the Synchrony Fellowship. Nima was also partially supported by the NSERC Discovery Program (Grant No. N01035) during his preliminary research on unlearning privacy in Canada, prior to beginning his Ph.D. at UConn.

Ethics Considerations

In accordance with the USENIX Security ethics guidelines, we have thoroughly considered the ethical implications of our work. RULI is a framework designed to evaluate the privacy risks and efficacy of unlearning methods. It alerts researchers and practitioners to carefully evaluate unlearning methods that promote the development of more effective unlearning strategies. By exposing empirical risks, RULI raises awareness of potential privacy issues that may be overlooked in current unlearning methods before deployment in practice.

We acknowledge the potential for misuse, as RULI (its attack part) could be employed by malicious actors to attack real-world models that claim to have unlearned specific data. However, we believe the research benefits outweigh these risks. Our findings offer practical mitigation strategies (in the Appendix). One approach is to improve unlearning designs by accounting for variation in memorization and privacy risks across samples, rather than treating all samples equally. This may require estimating and ranking the privacy risk of individual samples, which, while potentially costly, is feasible in systems designed with privacy in mind. Another mitigation strategy is to restrict the types of queries users can perform, such as limiting access to logit outputs or confidence scores, to reduce the potential for exploitation.

All experiments in our study were conducted using open-source datasets and models. The privacy attacks were simulated solely for the purpose of evaluation, and no real-world, confidential, or sensitive data were used. No individuals or organizations are impacted by these experiments. As such, Institutional Review Board (IRB) approval was not required.

Open Science

We adhered to all applicable guidelines relevant to our research domain, including compliance with open science policies. Our research artifacts are available and regularly maintained on GitHub[4]. The artifacts are also archived on Zenodo: https://zenodo.org/records/15610191. The archive (`ruli.zip`) includes: **(1) core.attack** – a complete implementation of the RULI framework, including modules for target sample preparation, model training, unlearning methods, shadow model construction, and evaluation; **(2) core.examples** – ready-to-run scripts and examples for end-to-end execution of target sample injection and inference; **(3) text** – an extension of our attack adapted for language model unlearning of 7-gram sequences.

[4] https://github.com/datasec-lab/Ruli

References

[1] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), 2016. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679.

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *CCS*, pages 308–318, 2016.

[3] Michael Aerni, Jie Zhang, and Florian Tramèr. Evaluations of machine learning privacy defenses are misleading. In *CCS*, pages 1271–1284, 2024.

[4] Meenatchi Sundaram Muthu Selva Annamalai, Georgi Ganev, and Emiliano De Cristofaro. "what do you want from theory alone?" experimenting with tight auditing of differentially private synthetic data generation. In *USENIX Security*, 2024.

[5] Caridad Arroyo Arevalo, Sayedeh Leila Noorbakhsh, Yun Dong, Yuan Hong, and Binghui Wang. Task-agnostic privacy-preserving representation learning for federated learning against attribute inference attacks. In *AAAI*, 2024.

[6] Thomas Baumhauer, Pascal Schöttle, and Matthias Zeppelzauer. Machine unlearning: Linear filtration for logit-based classifiers. *Machine Learning*, 111(9):3203–3226, 2022.

[7] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, 2021.

[8] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *ICML*, pages 2397–2430. PMLR, 2023.

[9] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *S&P*, pages 141–159. IEEE, 2021.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020.

[11] California State Legislature. California Consumer Privacy Act of 2018, 6 2018. Cal. Civ. Code § 1798.100 et seq. URL: https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375.

[12] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *S&P*, pages 1897–1914, 2022.

[13] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative. In *NeurIPS*, 2022.

[14] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security*, pages 267–284, 2019.

[15] Dingfan Chen, Ning Yu, and Mario Fritz. Relaxloss: Defending membership inference attacks without losing utility. In *ICLR*, 2022.

[16] Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *CVPR*, 2023.

[17] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When machine unlearning jeopardizes privacy. In *CCS*, pages 896–911, 2021.

[18] Eli Chien, Haoyu Wang, Ziang Chen, and Pan Li. Stochastic gradient langevin unlearning. *arXiv preprint arXiv:2403.17105*, 2024.

[19] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *ICML*, pages 1964–1974, 2021.

[20] Rishav Chourasia and Neil Shah. Forget unlearning: Towards true data-deletion in machine learning. In *ICML*, 2023.

[21] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *AAAI*, 2023.

[22] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 2023.

[23] Debeshee Das, Jie Zhang, and Florian Tramèr. Blind baselines beat membership inference attacks for foundation models. *arXiv preprint arXiv:2406.16201*, 2024.

[24] Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? In *COLM*, 2024.

[25] Chongyu Fan, Jiancheng Liu, Alfred Hero, and Sijia Liu. Challenging forgets: Unveiling the worst-case forget sets in machine unlearning. In *ECCV*, pages 278–297, 2025.

[26] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*, 2023.

[27] Shuya Feng, Meisam Mohammady, Hanbin Hong, Shenao Yan, Ashish Kundu, Binghui Wang, and Yuan Hong. Harmonizing differential privacy mechanisms for federated learning: Boosting accuracy and convergence. In *CODASPY*, 2025.

[28] Shuya Feng, Meisam Mohammady, Han Wang, Xiaochen Li, Zhan Qin, and Yuan Hong. DPI: ensuring strict differential privacy for infinite data streaming. In *S&P*, 2024.

[29] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *CCS*, pages 1322–1333, 2015.

[30] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *CCS*, pages 619–633, 2018.

[31] Ji Gao, Sanjam Garg, Mohammad Mahmoody, and Prashant Nalini Vasudevan. Deletion inference, reconstruction, and compliance in machine (un) learning. In *PETS*, 2022.

[32] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *NeurIPS*, 2019.

[33] Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *CVPR*, pages 792–801, 2021.

[34] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *CVPR*, pages 9304–9312, 2020.

[35] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *AAAI*, 2021.

[36] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *ICML*, 2020.

[37] Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. *NeurIPS*, 34:16319–16330, 2021.

[38] Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. Inexact unlearning needs more careful evaluations to avoid a false sense of privacy. In *SaTML*, 2025.

[39] Hongsheng Hu, Shuo Wang, Tian Dong, and Minhui Xue. Learn what you want to unlearn: Unlearning inversion attacks against machine unlearning. In *S&P*, 2024.

[40] Qiang Hu, Hengxiang Zhang, and Hongxin Wei. Defending membership inference attacks via privacy-aware sparsity tuning. *arXiv preprint arXiv:2410.06814*, 2024.

[41] Yangsibo Huang, Daogao Liu, Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Milad Nasr, Amer Sinha, and Chiyuan Zhang. Unlearn and burn: Adversarial machine unlearning requests destroy model accuracy. In *ICLR*, 2025.

[42] Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, et al. Measuring forgetting of memorized training examples. In *ICLR*, 2023.

[43] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *ACL*, pages 14389–14408, Toronto, Canada, July 2023.

[44] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in Neural Information Processing Systems*, 36, 2024.

[45] Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pages 243–254. PMLR, 2022.

[46] Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, Sijia Liu, et al. Model sparsity can simplify machine unlearning. *NeurIPS*, 36, 2024.

[47] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.

[48] Zhuo Ma, Yang Liu, Ximeng Liu, Jian Liu, Jianfeng Ma, and Kui Ren. Learn to forget: Machine unlearning via neuron masking. *TDSC*, 20(4):3194–3207, 2022.

[49] Saeed Mahloujifar, Huseyin A Inan, Melissa Chase, Esha Ghosh, and Marcello Hasegawa. Membership inference on word embedding and beyond. *arXiv preprint arXiv:2106.11384*, 2021.

[50] James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

[51] Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. Can LLMs keep a secret? testing privacy implications of language models via contextual integrity theory. In *ICLR*, 2024.

[52] Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. Tight auditing of differentially private machine learning. In *USENIX Security*, pages 1631–1648, 2023.

[53] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *CCS*, pages 634–646, 2018.

[54] Sayedeh Leila Noorbakhsh, Binghui Zhang, Yuan Hong, and Binghui Wang. Inf2Guard: An Information-Theoretic framework for learning Privacy-Preserving representations against inference attacks. In *USENIX Security*, 2024.

[55] Ashwinee Panda, Xinyu Tang, Christopher A. Choquette-Choo, Milad Nasr, and Prateek Mittal. Privacy auditing of large language models. In *ICLR*, 2025.

[56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[57] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[58] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[59] Ahmed Salem, Giovanni Cherubin, David Evans, Boris Köpf, Andrew Paverd, Anshuman Suri, Shruti Tople, and Santiago Zanella-Béguelin. Sok: Let the privacy games begin! a unified treatment of data inference privacy in machine learning. In *S&P*, 2023.

[60] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *NeurIPS*, 34:18075–18086, 2021.

[61] Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. In *ICLR*, 2025.

[62] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *S&P*, pages 3–18. IEEE, 2017.

[63] Ilia Shumailov, Jamie Hayes, Eleni Triantafillou, Guillermo Ortiz-Jimenez, Nicolas Papernot, Matthew Jagielski, Itay Yona, Heidi Howard, and Eugene Bagdasaryan. Ununlearning: Unlearning is not sufficient for content regulation in advanced generative ai. *arXiv preprint arXiv:2407.00106*, 2024.

[64] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[65] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security*, 2021.

[66] Thomas Steinke, Milad Nasr, and Matthew Jagielski. Privacy auditing with one (1) training run. *NeurIPS*, 2024.

[67] Joshua Stock, Jens Wettlaufer, Daniel Demmler, and Hannes Federrath. Lessons learned: defending against property inference attacks. *arXiv preprint arXiv:2205.08821*, 2022.

[68] Xinyu Tang, Saeed Mahloujifar, Liwei Song, Virat Shejwalkar, Milad Nasr, Amir Houmansadr, and Prateek Mittal. Mitigating membership inference attacks by self-distillation through a novel ensemble architecture. In *USENIX Security*, 2022.

[69] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Trans. Neural Netw. Learn. Syst*, 2023.

[70] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *EuroS&P*, pages 303–319. IEEE, 2022.

[71] Anvith Thudi, Hengrui Jia, Casey Meehan, Ilia Shumailov, and Nicolas Papernot. Gradients look alike: Sensitivity is often overestimated in DP-SGD. In *USENIX Security*, 2024.

[72] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *USENIX Security*, 2022.

[73] Florian Tramèr, Reza Shokri, Ayrton San Joaquin, Hoang Le, Matthew Jagielski, Sanghyun Hong, and Nicholas Carlini. Truth serum: Poisoning machine learning models to reveal their secrets. In *CCS*, pages 2779–2792, 2022.

[74] Eleni Triantafillou, Peter Kairouz, Fabian Pedregosa, Jamie Hayes, Meghdad Kurmanji, Kairan Zhao, Vincent Dumoulin, Julio Jacques Junior, Ioannis Mitliagkas, Jun Wan, et al. Are we making progress in unlearning? findings from the first neurips unlearning competition. *arXiv:2406.09073*, 2024.

[75] Yuxin Wen, Leo Marchyok, Sanghyun Hong, Jonas Geiping, Tom Goldstein, and Nicholas Carlini. Privacy backdoors: Enhancing membership inference through poisoning pre-trained models. In *NeurIPS*, 2024.

[76] Leon Wichert and Sandipan Sikdar. Rethinking evaluation methods for machine unlearning. In *EMNLP Findings*, 2024.

[77] Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. Machine unlearning: A survey. *ACM Comput. Surv.*, 56(1), aug 2023.

[78] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *CSF*, pages 268–282. IEEE, 2018.

[79] Xiaoyong Yuan and Lan Zhang. Membership inference attacks and defenses in neural network pruning. In *USENIX Security*, pages 4561–4578, 2022.

[80] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In *ICML*, 2024.

[81] Binchi Zhang, Yushun Dong, Tianhao Wang, and Jundong Li. Towards certified unlearning for deep neural networks. In *ICML*, 2024.

[82] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

Appendix

A Further Implementation Details

A.1 Training Details

The training configurations for different models and datasets involved in the experiments are shown in Table 7. All models were developed using PyTorch [56] for image classification; for text generation, we used Hugging Face [5] implementations of the language models for training and unlearning. The experiments are performed on servers equipped with multiple NVIDIA H100 Hopper GPUs (80 GB each).

[5] Pythia on Hugging Face, GPT-2 on Hugging Face

Table 6: Frequently used notations.

Symbol	Meaning
θ_{I}	Trained model
θ^*	Intermediate model parameter after t steps unlearning optimizations
$\theta_{\mathcal{R}}$	Retrained model
$\theta_{\mathcal{U}}$	Unlearned model
$\theta_{\mathcal{T}}$	Test model
D_{train}	Training data
$\mathcal{D}$	Dataset Distribtion
D_r	Remain data
D_f	Forget data
D_{target}	Target data
z	Target sample
$\mathcal{Q}_h(z)$	held-out distribution (from a sample was not in training and unlearning
$\mathcal{Q}_r(z)$	remained distribution (from a sample was in training but not in unlearning)
$\mathcal{Q}_u(z)$	unlearned distribution (from a sample was in training and in unlearning)
$\mathcal{Q}_i(z)$	in distribution (from a sample was in training)
$\mathcal{Q}_R(z)$	retrain distribution (from a sample retrained/not in training)
$\mathcal{A}$	Training algorithm
$\mathcal{U}$	Unlearning algorithm
A	Adversary (attacker)
C	Challenger
N	Number of shadow models

Table 7: Training configurations for different models/datasets

Training Parameters	**CIFAR-10**	**CIFAR-100**	**TinyImageNet**
	ResNet-18	**ResNet-18**	**Swin-small**
Training/fine-tuning epochs	50	50	2
Batch size	128	128	128
Momentum	0.9	0.9	0.9
Learning rate	1×10^{-1}	1×10^{-1}	1×10^{-4}
Optimizer	SGD	SGD	AdamW
ℓ_2 regularization	5×10^{-4}	5×10^{-4}	5×10^{-4}

A.2 Hyperparameters of Unlearning Baselines

We carefully tuned the hyperparameters to ensure that the unlearning baselines are close to the accuracy performance of the Retrain gold standard. We assumed the model provider can adapt the parameters adaptively according to the forget data (though in practice, this might be challenging). Specifically, we conducted a comprehensive grid search, tuning the parameter sets to each unlearning setting. Table 9 presents the hyperparameters used for various unlearning settings.

A.3 Target Data

Details on Identifying Vulnerable and Protected Samples. We train 256 shadow models on the dataset and set TPR@0.01% FPR as the threshold for classifying highly memorized samples. In this scenario, the attacker's confidence is quantified by the likelihood ratio ($\tau = \frac{p(In)}{p(In)+p(Out)}$), which, if it hovers around 0.5 ± 10^{-3}, approximates a coin flip in predicting membership. Our results, detailed in Table 10, indicate that 627 samples (approximately 2.5%) in CIFAR-10 can be chosen as vulnerable samples. We noticed most of

Table 8: Training configurations for language model unlearning using SFT with prefix language modeling.

Training Parameters	**Configuration**
Supervised training epochs	5
Batch size	16
Learning rate	5×10^{-5}
Optimizer	AdamW
Prefix language modeling	Enabled (+2 epochs)
Early Stopping	Enabled
Weight decay	0.01

Table 9: Hyper-parameters in unlearning benchmarks.

	Unlearning Tunable Hyper-parameters
ℓ_1-sparse	learning rate: {0.01}, sparsity parameter (α): $\{10^{-4}, 2\times10^{-4}, 5\times10^{-4}, 3\times10^{-4}\}$, batch size: {16, 64, 128}, sparsity scheduler: {linear increase, decay, constant}
Scrub	learning rate: $\{5\times10^{-4}, 10^{-4}, 5\times10^{-5}\}$, α: 0.1, β: 0.0, γ: 0.99, forget batch size: {16, 64, 128}, retain batch size: {16, 64, 128}, maximizing steps: {1, 2, 4, 6}, minimizing steps: {2, 5, 6, 10}
GA+	learning rate: $\{10^{-4}, 10^{-3}\}$, unlearn epochs: {3, 5, 7, 10, 15, 20}, refine epochs: {0, 3, 4, 5}, batch size: {16, 64, 128}
NegGrad+	learning rate: 5×10^{-4}, 5×10^{-5}, α: {0.1, 0.3, 0.35, 0.4, 0.5, 0.9}, forget batch size: {16, 64, 128}, retain batch size: {16, 64, 128}, maximizing steps: {1, 2, 4, 6}

the remaining samples (16,586 out of 25,000 samples) are less memorized and difficult for an attacker to infer membership. We consider data samples other than these 627 samples as safe and protected ones. As expected, 2,583 samples (approximately 10.33%) in CIFAR-100 are highly memorized. Conversely, we identified 16,586 less memorized samples in CIFAR-10 and 6,639 in CIFAR-100.

Table 10: MIA's TPR@FPR and number of vulnerable samples (256 shadow models on one-round membership inference).

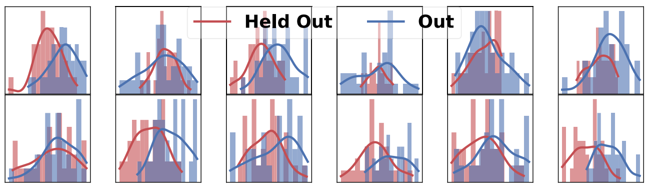
Dataset	TPR@ 0.1%FPR	TPR@ 0.01%FPR	#Highly memorized	#Less memorized
CIFAR-10	4.1%	2.5%	627	16,586
CIFAR-100	24.33%	10.33%	2,583	6,639

A.3.1 Target Data Indistinguishability

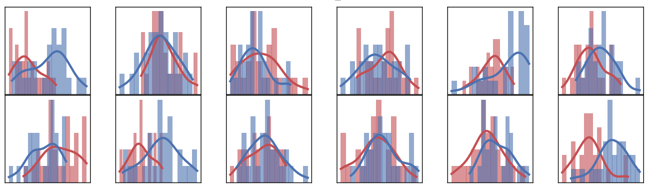
Image. To ensure the reliability of membership inference evaluations, it is critical that the selected two cases of target data do not exhibit distribution shifts and are resistant to *blind attacks* [23]. In blind attacks, the adversary does not rely on model training and instead attempts to distinguish data solely based on its distributional characteristics. We assess the indistinguishability of members (*Unlearned*) and non-members (*Out/Held-out*) samples using two metrics: Label Overlap, which quantifies the proportion of shared classes to rule out label-based separability; and Embedding Classifier, which trains a linear classifer from embeddings extracted from a ViT-based model to distinguish the two sets. A near-chance classifer accuracy (≈50%) indicates no clear distributional difference. We also include a uniformly random selection baseline to reflect the expected distinguishability when selecting 250 samples from each set—particularly. Results in Table 11 show both canary (our targets) and random selections yield similar overlap and indistinguishability.

Table 11: Blind attack results: label overlap and classifier accuracy.

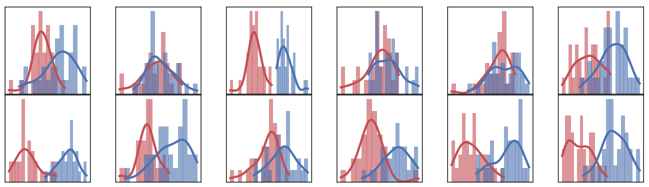
	Canary (ours)		Random	
	Labels Overlap	Accuracy (Classifier)	Labels Overlap	Accuracy (Classifier)
CIFAR-10	92.0%	52.0%	91.6%	51.3%
CIFAR-100	87.2%	51.3%	87.6%	50.0%
TinyImageNet	54.8%	53.3%	53.2%	46.7%



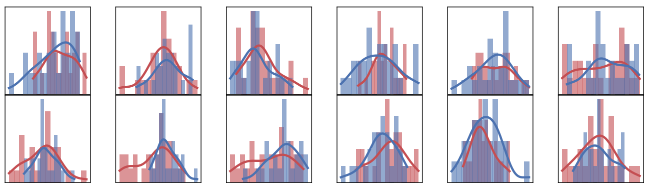
(a) ℓ_1 Sparse



(b) Scrub



(c) GA+



(d) Retrain

Figure 9: Logit-scaled confidences of two distributions held-out (Q_h) and out (Q_R) from 16 randomly samples of class-5 of CIFAR-10 (compatible with the U-LiRA setting). Our empirical evidences are showing that Q_h is not similar to Q_u unless in Retraining.

Text. We randomly select the *out* and *unlearned* target sequences from the test set of WikiText-103, ensuring that both groups are drawn from the same underlying distribution. Since our membership inference only queries the final

7 tokens of each sample, we evaluate whether these target spans are lexically indistinguishable from one another. We compute both exact 7-gram matches and token-wise (1-gram) overlap between the out and unlearned spans under different selection modes: first 7 tokens, final 7 tokens, and random 7 tokens. Exact matches are rare in all settings ($\leq 2\%$), but we observe relatively high token-wise overlap for the final 7 tokens (mean: 54.3%), which we use as our targets.

B Empirical evidence of mismatch between *Out* and *Held-out* distributions

To demonstrate the distinguishability of Q_h and Q_R, and to support our claim in Section 4.3.2 that these two distributions are not similar for all samples, we present logit-scaled confidences derived from our shadow models representing these distributions. We selected 12 random samples from class 5 of the CIFAR-10 dataset and utilized 90 shadow models for this analysis. The logit-scaled confidences were evaluated using ℓ_1 Sparse, GA+, NegGrad+, and Retrain (standard) benchmarks, as shown in Figure 9. In the figure, the red distribution represents cases where the sample was excluded from both shadow model training and shadow model unlearning (held-out), while the blue distribution represents cases where the sample was excluded from training alone (out). Our results demonstrate that for inexact unlearning benchmarks, such as ℓ_1 Sparse, GA+, and NegGrad+, the held-out and out distributions differ significantly across all 12 samples. In contrast, when employing the Retrain (standard) benchmark, these distributions appear similar, suggesting consistency. This discrepancy highlights a key limitation of inexact unlearning methods, where $Q_h \not\sim Q_R$, directly challenges the assumption that these distributions are interchangeable. This finding emphasizes the need for more nuanced evaluation frameworks to account for such distributional differences in unlearning.

C Discussions

C.1 Possible Solutions to Mitigate Privacy Risk of Inexact Unlearning Designs

We discuss two possible solutions to mitigate privacy leakage of inexact unlearning.

Estimating Memorization. To mitigate the negative effects of batch optimizations during unlearning, one approach is to estimate the memorization levels of all forget samples and sequentially unlearn similar samples within the same round. However, accurately assessing per-sample memorization is computationally expensive, often requiring the training of numerous models. Prior work has shown that many of the most vulnerable samples tend to be mislabeled or ambiguous, suggesting that model trainers could use static analysis techniques to pre-identify and handle these cases separately. As a proof of concept, we assume that the model provider has some means of estimating per-sample memorization and can construct mini-batches ranked by the memorization scores of the forget data. Using this strategy, we observe a notable reduction in attack performance: attack accuracy drops to 61%, and TPR@FPR 1% decreases to 3% on the Scrub unlearning.

Training with DP-SGD. Differentially Private SGD (DP-SGD) [2] is a principled training framework that provides worst-case privacy guarantees for every individual sample. This enables provable, per-sample privacy estimates [71], which can be leveraged to assess the privacy status of forget samples. Such a property makes DP-trained models a compelling candidate for unlearning, as one can, in theory, show that the effective privacy loss (ε) for a forgotten sample is near zero. However, DP training is not commonly adopted in practice due to its privacy-utility trade-offs [28]. Despite this, designing unlearning algorithms specifically for DP-trained models remains a promising direction. Since DP-trained models are often meant to protect sensitive data, effective unlearning in this setting could be both feasible and beneficial. Exploring unlearning techniques that push the per-sample ε lower (ideally towards zero) could offer a new perspective on practical, certifiable unlearning.

C.2 Notes on Provided Results

Sensitivity to Hyperparameters. We observe that unlearning algorithms are significantly more sensitive to hyperparameter settings than standard training, especially when forgetting high-risk canaries. While we perform grid search to select reasonable settings, small variations can lead to meaningful shifts in attack performance. For example, applying the CIFAR-10 unlearning setting to CIFAR-100—by reducing just one unlearning step—raises the TPR@1%FPR to 24.5%. This suggests that even subtle changes can disturb unlearning performance and even possibly amplify privacy leakage.

Varying Canary Rates. One can vary the canary rates without retraining the shadow models. In our default setting (Section 6.2.2), nearly 55% of the forget set consists of vulnerable samples canaries. As an illustrative example, we reduce this rate to 5% by injecting more non-canary samples into forget data, and adjust the Scrub unlearning hyperparameters accordingly. The attack remains effective. On CIFAR-10, we observe an average TPR@1%FPR of 8.7%, TPR@5%FPR of 26.6%, and attack accuracy of 68.5%. On TinyImageNet, TPR@1%FPR reaches 9.6%, TPR@5%FPR is 28.2%, and accuracy is 65.2%. These results demonstrate that RULI's effectiveness at a lower canary rate.