RETRACE: REINFORCEMENT LEARNING-GUIDED RECONSTRUCTION ATTACKS ON MACHINE UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning has emerged as an inevitable AI mechanism to support GDPR requirements such as revoking user consent through the "right to be forgotten". However, existing approaches often leave residual traces that make them vulnerable to data reconstruction attacks. In this work, we propose RETRACE, the *first reconstruction attack framework* that uniquely formulates unlearned data recovery on *large-scale deep architectures* as a reinforcement learning (RL) problem. By treating residual unlearning *traces* as reward signals, RETRACE guides a generator to actively explore the input space and converge toward the forgotten data distribution. This RL-guided approach enables both *instance-level* recovery of individual samples and *distribution-level* reconstruction of unlearned classes. We provide a theoretical foundation showing that the RL objective converges to an exponential-tilted distribution that amplifies forgotten regions. Empirically, RETRACE achieves up to 73.1% instance-level recovery and reduces FID and KL scores beyond state-of-the-art baselines, UIA (IEEE S&P 2024) and HRec (NeurIPS 2024). Strikingly, on the challenging task of text unlearning, it improves BLEU scores by nearly 100% over black-box baselines while preserving distributional fidelity, demonstrating that RL can recover even high-dimensional and structured modalities. Furthermore, RETRACE demonstrates effectiveness across both convolutional (ResNet) and transformer-based models, with DistilBERT as the largest architecture attacked to date. These results show that current unlearning methods remain vulnerable, highlighting the need for robust and provably private mechanisms.

1 INTRODUCTION

Machine unlearning has recently emerged as a critical technique to empower users with the ability to remove their data from trained models, aligning with the "right to be forgotten" in GDPR (European Parliament & Council of the European Union, 2016) and responsible Artificial Intelligence (AI) innovation. To achieve this, early work on exact unlearning enforces deletion by retraining the model from scratch on the dataset with the target samples removed. While this approach provides a strong guarantee, it is computationally prohibitive for today's large and complex models (Ginart et al., 2019). As a result, recent research has shifted toward *approximate unlearning* (Bourtoule et al., 2021; Li et al., 2024), which directly modifies a trained model to erase the influence of unlearned data. These techniques offer more practical trade-offs between efficiency and unlearned strength, making machine unlearning a cornerstone of responsible AI innovation, where compliance with user rights and trustworthy data governance is essential.

Despite its promise, machine unlearning is vulnerable to *reconstruction attacks* (Zhang et al., 2023; Bertran et al., 2024), where adversaries attempt to recover the data that was intended to be forgotten, as unlearning may have the unintended opposite effect: rather than concealing the data, it can inadvertently facilitate the localization of sensitive records in a vast corpus. As a result, it is easier for adversaries to search the needle in the haystack. Such attacks pose a direct threat to data privacy, as they might effectively reverse the unlearning process and expose sensitive information. The consequences can be severe in domains such as healthcare. For example, in a medical setting, even if a patient requests deletion of their medical images or health records, a reconstruction attack could still recover identifiable details and compromise confidentiality. Such risk highlights the urgent need to systematically investigate the vulnerability of unlearning mechanisms under reconstruction attacks.

Several recent studies (Bertran et al., 2024; Pang et al., 2024; Hu et al., 2024) have explored reconstruction under unlearning. One line relies on *closed-form parameter analysis* (Bertran et al., 2024), which can exactly recover deleted samples but only applies to simple or linear models. Another line, *update-based reconstruction* (Hu et al., 2024), requires white-box gradients or parameter updates and is restricted to instance-level recovery, failing to generalize when multiple deletions occur. These limitations motivate the need for a framework that exploits unlearning traces in deep models, supports diverse access levels, and enables both instance- and distribution-level recovery.

Our work. In this work, we propose RETRACE, the first reconstruction attack framework that exposes privacy vulnerabilities in machine unlearning on both convolutional (ResNet He et al. (2016)) and transformer-based (Distil-BERT Sanh et al. (2019)) models, with Distil-BERT being the largest-scale architecture attacked to date. The key insight is that unlearning leaves detectable *traces* between pre- and post-unlearning models (Chen et al., 2025), which can be exploited with *reinforcement learning (RL)*. RETRACE operates in three steps. The first is *trace extraction*, which integrates signals such as output shifts, loss differences, and gradient alignments across model access levels, i.e., black-, grey-, and white-box. The second is *RL-guided reconstruction*, where a generator explores the input space and optimizes trace scores as rewards, enabling recovery even from complex models. The third is candidate selection and refinement, which supports both *instance-level* and *distribution-level* recovery.

We evaluate RETRACE through both theoretical analysis and empirical validation. On the theoretical side, we first establish that the RL objective converges to an exponential-tilted distribution, which provably amplifies regions with stronger unlearning traces. Building on this characterization, we show that RETRACE can successfully recover unlearned data at the instance level with high probability once a sufficient number of candidates are generated. Furthermore, using a bias–variance decomposition, we demonstrate that the empirical distribution of reconstructed samples converges toward the deleted-data distribution, with the bias controlled by the separability margin and the variance diminishing with the number of samples.

On the experimental side, we first visualize unlearning traces across different access levels and confirm that unlearned data leave consistent, instance-aligned residual signals. We then evaluate RETRACE on three benchmarking datasets under both exact and approximate unlearning. At the instance level, RETRACE can achieve the best success rate of 73.1% with an MSE of 0.17, demonstrating its ability to generate faithful reconstructions. At the distribution level, it can reduce the Fréchet Inception Distance (FID) to 99.1 and the Kullback-Leibler (KL) divergence to 2.53, showing strong alignment with the unlearned data distribution. Compared with baseline methods, RETRACE consistently attains lower MSE, higher success rates, and stronger feature similarity, confirming its superior effectiveness. Ablation study further demonstrates its robustness and generalization.

Contributions. Our main contributions are summarized as follows.

- **An impactful vulnerability unveiled on machine unlearning**. Most current unlearning techniques do not completely erase the influence of deleted data but instead leave discernible traces in the model. We find that these residual traces can be leveraged as reliable signals to reconstruct the supposedly unlearned data.
- **A novel reconstruction attack framework**. We propose RETRACE, a trace-guided RL-based reconstruction attack that systematically exploits unlearning-induced traces to recover unlearned data at both the instance and distribution levels. Different model access levels further make RETRACE more practical.
- **A comprehensive study and evaluation**. We provide a theoretical foundation for RETRACE in reconstructing the unlearned data at an instance level and distribution level, demonstrating its effectiveness. We also conduct comprehensive experimental evaluations to further show its performance.

2 PROBLEM STATEMENT

2.1 SYSTEM SETTING

Unlearning settings. We consider the standard machine unlearning setting. Let $\mathcal{D} = \mathcal{D}_{\text{ret}} \cup \mathcal{D}_{\text{del}}$ denote the training dataset, where $\mathcal{D}_{\text{ret}}$ represents the samples to be retained and $\mathcal{D}_{\text{del}}$ represents

the samples requested to be deleted. A model is first trained on the full dataset $\mathcal{D}$, resulting in the *pre-unlearning model* f^+ with parameters θ^+. Upon receiving a deletion request, an unlearning algorithm is applied to remove the influence of $\mathcal{D}_{\text{del}}$, producing the *post-unlearning model* f^- with parameters θ^-. The objective of unlearning is to ensure that f^- behaves as if it were trained only on $\mathcal{D}_{\text{ret}}$, thereby eliminating any contribution of $\mathcal{D}_{\text{del}}$ to the model.

RETRACE position. We position RETRACE as a new form of reconstruction attack against machine unlearning. By leveraging RL, it progressively reconstructs the forgotten data (either at the instance level or distribution level) from residual traces left after unlearning.

2.2 THREAT MODEL

Adversary knowledge. We assume the adversary has access to both the pre-unlearning model f^+ and the post-unlearning model f^-, a scenario that arises in practice through model versioning, API updates, or common deployment practices (see Appendix A for evidence). Depending on the deployment, we consider three levels of model access. ***Black-box access:*** The adversary can query f^+ and f^- and obtain their prediction outputs (e.g., labels and prediction probabilities). ***Grey-box access:*** In addition to outputs, the adversary can compute the loss with respect to the task loss function. ***White-box access:*** The adversary has full knowledge of model parameters θ^+ and θ^-, so that they can also obtain the gradient in training.

The adversary does *not* have access to the original training dataset $\mathcal{D}$ or the forgotten set $\mathcal{D}_{\text{del}}$. Instead, they may rely on an auxiliary public dataset $\mathcal{D}_{\text{pub}}$ drawn from the same distribution, which can be used to initialize candidate inputs for trace extraction.

Attack goal. The adversary's goal is to reconstruct the data belonging to the forgotten class. We consider two levels of reconstruction: (i) *instance-level*, where the adversary attempts to recover an individual sample in the unlearned class; and (ii) *distribution-level*, where the adversary aims to approximate the overall distribution of $\mathcal{D}_{\text{del}}$. Both types of recovery undermine the privacy guarantees of unlearning and expose sensitive information that should have been removed. Formally, the goal can be expressed as follows.

Instance-level reconstruction. Let $d_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a task-appropriate metric. Given $\varepsilon > 0$, we say an adversary $\mathcal{A}$ succeeds at instance-level reconstruction if it outputs $\widehat{x} \in \mathcal{X}$ such that

$$\exists\,(x, y) \in \mathcal{D}_{\text{del}} \;\text{ with }\; d_{\mathcal{X}}(\widehat{x}, x) \leq \varepsilon. \tag{1}$$

For a k-set $\widehat{X} = \{\widehat{x}_1, \ldots, \widehat{x}_k\}$, success can be measured by the minimum matching distance (MMD) $\text{MMDist}(\widehat{X}, \mathcal{D}_{\text{del}}) = \min_\pi \frac{1}{k} \sum_{j=1}^{k} d_{\mathcal{X}}(\widehat{x}_j, x_{\pi(j)})$ and a threshold ε.

Distribution-level reconstruction. Let $\mathbb{P}_{\text{del}}$ denote the (unknown) distribution of inputs in the deleted set $\mathcal{D}_{\text{del}}$, and let $\widehat{\mathbb{P}}$ be the distribution induced by the reconstructed samples. We say the adversary achieves distribution-level reconstruction if the reconstructed distribution $\widehat{\mathbb{P}}$ is close to $\mathbb{P}_{\text{del}}$ under some statistical distance $d(\cdot, \cdot)$, i.e., $d(\widehat{\mathbb{P}}, \mathbb{P}_{\text{del}}) \leq \varepsilon$, for a small threshold $\varepsilon > 0$.

3 APPROACH

In this section, we delve into the internals of RETRACE. It is designed to reconstruct unlearned data by exploiting residual traces left after unlearning. Our insight is that when a model undergoes unlearning, the discrepancy between the pre-unlearning model f^+ and the post-unlearning model f^- inevitably reveals subtle *unlearning traces* (Chen et al., 2025). These traces provide exploitable *signals* regarding the unlearned data. To leverage these signals effectively, we formulate the reconstruction problem as a *reinforcement learning (RL)* task. The learning agent explores the input space, and the feedback it receives, derived from unlearning traces, serves as the reward. By optimizing toward higher rewards, the agent gradually converges to regions of the input space that better approximate the unlearned data. The overall workflow of RETRACE is illustrated in Figure 1.

3.1 UNLEARNING TRACE EXTRACTION

The first step of RETRACE is to extract *unlearning traces*, which we define as *residual signals* that expose the behavioral differences between the pre-unlearning model f^+ and the post-unlearning

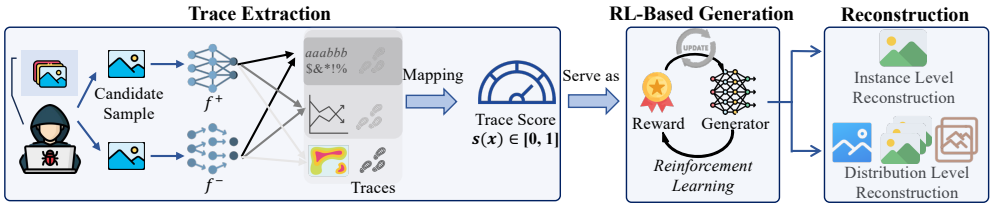


Figure 1: Overall workflow of RETRACE.

model f^-. Since such differences are only manifested when the models are queried with inputs, we probe the models with a candidate input x and extract the trace. Given x drawn from a public distribution $\mathcal{D}_{\text{pub}}$ or sampled from the generator π_ϕ, RETRACE quantifies traces at three progressively informative levels depending on the degree of model access. We follow (Sarmad et al., 2019) in designing trace extraction strategies tailored to different access levels.

Prediction-level traces. With black-box access, we measure the prediction discrepancy between f^+ and f^- to check whether unlearning leaves residual influence on the output distribution. The trace is defined as the ℓ_2 distance between their prediction vectors:

$$\delta_{\text{pred}}(x) = \|f^+(x) - f^-(x)\|_2. \tag{2}$$

Loss-level traces. With grey-box access, we extract traces from the task loss $\ell(\cdot)$. Using a pseudo-label $\hat{y}$ from f^+, we compute the absolute difference between the two models' losses:

$$\delta_{\text{loss}}(x) = \left|\ell(f^+(x), \hat{y}) - \ell(f^-(x), \hat{y})\right|. \tag{3}$$

Gradient-level traces. With white-box access, we capture differences in sensitivity to input perturbations. The trace is measured by the cosine distance between input gradients:

$$\delta_{\text{grad}}(x) = 1 - \cos\big(\nabla_x \ell(f^+(x), \hat{y}),\ \nabla_x \ell(f^-(x), \hat{y})\big). \tag{4}$$

Together, the unlearning trace of an input x under different access levels can be represented as:

$$T(x) = \begin{cases} \big(\delta_{\text{pred}}(x)\big), & \text{black-box access,} \\ \big(\delta_{\text{pred}}(x),\ \delta_{\text{loss}}(x)\big), & \text{grey-box access,} \\ \big(\delta_{\text{pred}}(x),\ \delta_{\text{loss}}(x),\ \delta_{\text{grad}}(x)\big), & \text{white-box access.} \end{cases} \tag{5}$$

$T(x)$ is then used to guide the unlearned data reconstruction process.

3.2 RL-GUIDED GENERATION

Formulation as RL. We cast reconstruction as an RL problem with four components. *State:* The current generation process, represented by the latent code z or partially generated sample x. *Action:* Extending or modifying the current state, e.g., sampling or refining a candidate x from the generator. *Policy:* The generator distribution π_ϕ parameterized by ϕ, which outputs candidates from the latent space. *Reward:* A weighted combination of unlearning traces that encourages π_ϕ to produce samples minimizing discrepancies between f^+ and f^-. Formally,

$$r(x) = \begin{cases} -\alpha\delta_{\text{pred}}(x), & \text{black-box,} \\ -\alpha\delta_{\text{pred}}(x) - \beta\delta_{\text{loss}}(x), & \text{grey-box,} \\ -\alpha\delta_{\text{pred}}(x) - \beta\delta_{\text{loss}}(x) - \gamma\delta_{\text{grad}}(x), & \text{white-box,} \end{cases} \tag{6}$$

with $\alpha, \beta, \gamma \geq 0$ controlling the relative weights.

In this formulation, the generator π_ϕ acts as the policy producing candidate reconstructions. The state is the current generation, actions modify it, and the environment queries (f^+, f^-) to obtain a trace score $s(x)$. For each sample $x \sim \pi_\phi$, the detector returns $s(x)$, which is used to update ϕ to maximize the expected score.

Objective of generator. The generator π_ϕ is optimized by maximizing

$$J(\phi) = \mathbb{E}_{x\sim\pi_\phi}[\,r(x)\,] \;-\; \lambda D(p_\phi \,\|\, \mathcal{D}_{\text{pub}}) \;+\; \tau\mathcal{H}(\pi_\phi), \tag{7}$$

where the trace score term drives reconstructions toward the unlearned distribution, the divergence penalty enforces realism by aligning p_ϕ with the public distribution $\mathcal{D}_{\text{pub}}$, and the entropy term $\mathcal{H}(\pi_\phi)$ encourages diversity. λ and τ balance these objectives.

Optimization. To optimize the generator policy π_ϕ, we adopt *Proximal Policy Optimization (PPO)* (Schulman et al., 2017), which stabilizes training compared to vanilla policy gradients by introducing a clipped surrogate objective:

$$L^{\text{PPO}}(\phi) = \mathbb{E}x \sim \pi\phi_{\text{old}}\Big[\min\big(r_\phi(x)A(x),;\text{clip}(r_\phi(x), 1-\epsilon, 1+\epsilon)A(x)\big)\Big], \tag{8}$$

where $r_\phi(x) = \pi_\phi(x)/\pi_{\phi_{\text{old}}}(x)$ is the importance ratio, $A(x)$ is the advantage from reward $r(x)$, and ϵ is a small trust-region parameter. This clipping prevents overly large updates, ensuring stable convergence while preserving exploration. By iteratively optimizing $L^{\text{PPO}}(\phi)$, the generator converges toward policies that produce high-trace, realistic, and diverse samples, effectively approximating the unlearned data distribution across access settings.

3.3 Reconstruction

After RL, the generator π_ϕ produces a set of candidate samples $\{x_i\}_{i=1}^N \sim \pi_\phi$. Each candidate is then evaluated by its trace score $s(x_i)$. The final reconstruction stage selects and refines the highest-scoring candidates as approximations of the unlearned data. This can be carried out at two levels.

Instance level. At the instance level, we first identify the top-scoring candidate $\hat{x}_0$:

$$\hat{x}_0 = \arg\max_{x_i} s(x_i), \tag{9}$$

which serves as the initial approximation of a forgotten instance. To obtain a more faithful reconstruction, we then refine $\hat{x}_0$ by solving the optimization:

$$\hat{x} = \arg\max_{x \in \mathcal{X}} \; s(x) - \gamma\Omega(x), \tag{10}$$

where the search space $\mathcal{X}$ is initialized around $\hat{x}_0$. Here $\Omega(x)$ enforces plausibility constraints such as pixel bounds for images, grammaticality for text, or statistical validity for structured data.

Distribution level. At the distribution level, we aim to approximate the entire unlearned data distribution. Specifically, we first rank all generated candidates by their trace scores $\{s(x_i)\}_{i=1}^N$ and select the top-k elements:

$$\widehat{\mathcal{D}}_{\text{forget}} = \{x_i \mid i \in I_k\}, \tag{11}$$

where I_k indexes the k highest-scoring candidates. Each element in $\widehat{\mathcal{D}}_{\text{forget}}$ is then refined in the same way as in the instance-level case, i.e., by optimizing $s(x) - \gamma\Omega(x)$ within its neighborhood.

3.4 Discussion on Efficiency

The overall complexity of ReTrace is $O(I \cdot N \cdot C_f + N \log N)$, dominated by generator sampling and trace evaluation across I RL iterations with N candidates per iteration. Here, C_f denotes the cost of a single model forward/backward pass. In practice, the runtime is comparable to standard adversarial training or RL-based generation pipelines. Detailed explanation on efficiency is in Appendix B.

4 Theoretical Analysis

In this section, we provide a theoretical foundation of ReTrace's attack effectiveness. Our goal is to formally establish that, under mild assumptions, RL optimization in ReTrace converges to policies that maximize unlearning traces, hence enabling effective reconstruction of the unlearned data distribution (see full proofs in Appendix C).

4.1 Definitions and Assumptions

Definition 1 (Trace score). *Let $s : \mathcal{X} \to [0, 1]$ denote the (fixed) trace score produced by the detector, quantifying the strength of unlearning traces for any $x \in \mathcal{X}$.*

Definition 2 (Trace separability). *Let $\mathbb{P}_{\mathrm{del}}$ and $\mathbb{P}_{\mathrm{ret}}$ denote the input distributions of the deleted and retained data. Traces are separable if there exists a margin $\Delta > 0$ such that*

$$\mathbb{E}_{x\sim\mathbb{P}_{\mathrm{del}}}[s(x)] \;\geq\; \mathbb{E}_{x\sim\mathbb{P}_{\mathrm{ret}}}[s(x)] + \Delta. \tag{12}$$

Assumption 1 (Policy expressiveness). *The policy class Π contains densities absolutely continuous w.r.t. a public prior p_0 that can approximate $\mathbb{P}_{\mathrm{del}}$ in total variation to arbitrary precision. In particular, Π includes the exponential-tilting family*

$$\pi_\lambda(x) \;\propto\; p_0(x)\exp(\lambda s(x)), \qquad \lambda \in \mathbb{R}. \tag{13}$$

Assumption 2 (RL convergence). *The generator is trained by entropy/KL-regularized RL with objective*

$$\mathcal{J}_\tau(\pi) = \mathbb{E}_{x\sim\pi}[s(x)] - \tau\,\mathrm{KL}(\pi\|p_0), \qquad \tau > 0, \tag{14}$$

and converges almost surely to a stationary point within Π.

Theorem 1 (Optimal policy form). *Under Assumptions 1 and 2, any stationary point $\pi^\star$ of $\mathcal{J}_\tau$ takes the form*

$$\pi^\star(x) \;\propto\; p_0(x)\exp\big(s(x)/\tau\big), \tag{15}$$

and is the unique maximizer within policy space Π.

Theorem 1 proves the exponential-tilted form $\pi^\star \propto p_0\, e^{s/\tau}$ by using a variational (Lagrangian) derivation under the normalization constraint together with the strict concavity of $\mathcal{J}_\tau$; the complete proof is provided in Appendix C.1.

Theorem 2 (Instance-level reconstruction). *Let $\pi^\star$ be as in Theorem 1 and fix $\varepsilon > 0$. Define*

$$p_\star := \pi^\star\Big(\mathcal{N}_\varepsilon(\mathrm{supp}(\mathbb{P}_{\mathrm{del}}))\Big). \tag{16}$$

Then for k i.i.d. candidates drawn from $\pi^\star$ and refined by the local step in §3.3,

$$\Pr\Big[\exists j \le k : \min_{x\in\mathcal{D}_{\mathrm{del}}} d_{\mathcal{X}}(\hat{x}_j, x) \le \varepsilon\Big] \;\geq\; 1 - (1-p_\star)^k. \tag{17}$$

Theorem 2 guarantees instance-level success by using the fact that $\pi^\star$ assigns positive probability mass to any ε-neighborhood of $\mathrm{supp}(\mathbb{P}_{\mathrm{del}})$ and a standard i.i.d. coverage bound $1-(1-p_\star)^k$; the full proof appears in Appendix C.2.

Theorem 3 (Distribution-level reconstruction). *Let $\widehat{P}_k$ be the empirical distribution of the reconstructed candidates. Under Definitions 1, 2 and Assumptions 1, 2, there exists a bias term $C_1(\tau,\Delta)$ such that, for sufficiently large k,*

$$d(\widehat{P}_k, \mathbb{P}_{\mathrm{del}}) \;\leq\; C_1(\tau,\Delta) \;+\; \epsilon(k,\delta), \tag{18}$$

with probability at least $1-\delta$. Here $d(\cdot,\cdot)$ is a general statistical distance (e.g., Maximum Mean Discrepancy (MMD), Wasserstein), and $\epsilon(k,\delta)$ is the sampling error vanishing as $k\to\infty$.

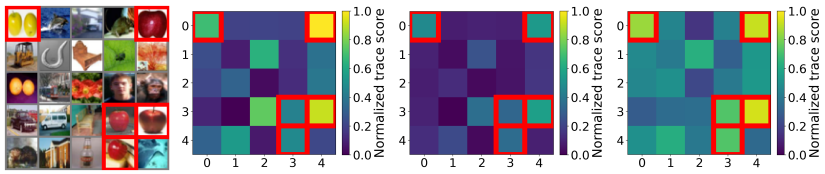
Theorem 3 establishes distribution-level reconstruction by using a bias–variance decomposition $d(\widehat{P}_k, \mathbb{P}_{\mathrm{del}}) \le d(\widehat{P}_k, \pi^\star) + d(\pi^\star, \mathbb{P}_{\mathrm{del}})$, where the bias stems from exponential tilting (controlled by (τ,Δ)) and the sampling term decays with k; the detailed argument is given in Appendix C.3.

5 Evaluation

5.1 Experimental Setup for Main Experiments

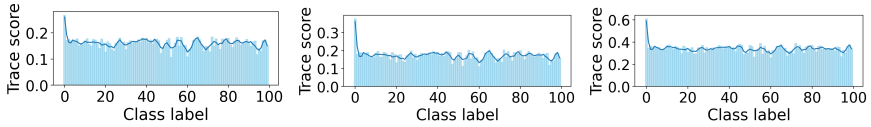
We evaluate ReTrace on three datasets: CIFAR-100 (Krizhevsky et al., 2009), Food-101 (Bossard et al., 2014), and PathMNIST (Yang et al., 2023), using ResNet-18 (He et al., 2016) classifiers with paired models (f^+, f^-) obtained by both exact and single gradient (approximate) unlearning (Thudi et al., 2022). Unlearning is performed in a class-wise manner with class 0 as the forgotten category. A DCGAN-style (Radford et al., 2015) generator is used for image tasks, guided by an RL policy to exploit unlearning traces across black-, grey-, and white-box access levels. We assess reconstruction quality at the instance level using *mean squared error (MSE)*, *cosine similarity (CS)* [1], and *success rate (SR)*, and at the distribution level using *Fréchet Inception Distance (FID)* and *Kullback–Leibler (KL) divergence*. For comparison, we include two state-of-the-art baselines: Unlearning Inversion Attack (UIA) (Hu et al., 2024) and HRec (Bertran et al., 2024). Detailed experimental settings are in Appendix D.1.

[1] We extract features from the pool3 layer of an ImageNet-pretrained Inception-V3 network to compute cosine similarity.



(a) Original (b) Black-box (c) Grey-box (d) White-box

Figure 2: Traces of unlearned data under different model access levels on CIFAR-100 in the approximate unlearning scenario. The images in red boxes represent the unlearned data "*Apple*".



(a) Black-box (b) Grey-box (c) White-box

Figure 3: Trace distribution under different model access levels on CIFAR-100 in the exact unlearning scenario. Class 0 represents "*Apple*".

5.2 Trace Visualization

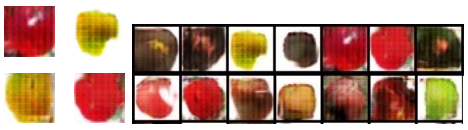
To empirically demonstrate the feasibility of using traces as reconstruction signals, we visualize the traces of unlearned data with heatmaps and further present their distributions, showing that the traces of unlearned data exhibit distinct patterns. Below, we explain the results on CIFAR-100 under approximate unlearning; the full evaluation results are provided in Appendix D.2.

Trace of unlearned data. Figure 2 visualizes unlearning traces on CIFAR-100 across three model-access settings. Panel (a) shows a 5×5 patch of original images with unlearned samples ("Apple") marked in red; Panels (b)–(d) present heatmaps of the normalized trace scores on the same grid. In the black-box setting, unlearned samples already exhibit top-quantile scores, showing that prediction-level discrepancies leave detectable signals. Grey-box access sharpens localization as loss-level differences enhance the contrast between forgotten and retained samples. White-box access further amplifies this effect: gradient-level information yields the clearest and most stable separation. Overall, unlearned data consistently leaves residual, instance-aligned traces across all access levels.

Trace distribution. We further analyze trace distributions across access levels. Figure 3 shows class-wise mean trace scores on CIFAR-100, with class 0 as the unlearned category "Apple". In the black-box setting, class 0 already scores higher than others, indicating measurable separation from prediction-level discrepancies. Grey-box access enlarges this gap by incorporating loss-level information, while white-box access further sharpens the contrast using gradient-level traces. Overall, the unlearned class consistently exhibits distinctly higher trace values, confirming that residual traces extend beyond instances to distributional patterns.

5.3 Effectiveness of ReTrace

We first benchmark the instance-level and distribution-level reconstruction capabilities of ReTrace under three model access settings across three benchmarking datasets, and then compare its performance with two state-of-the-art reconstruction attacks. Figure 4 provides some generated cases by ReTrace, and more cases are presented in Appendix D.3.



(a) Instance. (b) Distribution.

Figure 4: Generated unlearned samples ("Apple class") in CIFAR-100 under exact unlearning.

Benchmarking. For instance-level recovery, Table 1 shows that reconstructed samples achieve MSE values close to the intra-class criteria (e.g., CIFAR-100: 0.20 vs. 0.16), with SR around 50–60% under exact unlearning and further increasing under approximate unlearning (e.g., CIFAR-100:

Table 1: The effectiveness of RETRACE from instance-level reconstruction

Dataset	Exact Unlearning									Approximate Unlearning								
	Black-box			Grey-box			White-box			Black-box			Grey-box			White-box		
	MSE	SR	CS	MSE	SR	CS	MSE	SR	CS	MSE	SR	CS	MSE	SR	CS	MSE	SR	CS
CIFAR-100	0.23	52.3%	0.43	0.22	58.7%	0.46	0.20	62.4%	0.47	0.24	55.9%	0.46	0.21	68.7%	0.49	0.17	73.1%	0.50
Food-101	0.25	48.9%	0.38	0.23	54.7%	0.42	0.23	50.4%	0.44	0.20	56.6%	0.41	0.18	62.5%	0.49	0.16	65.3%	0.49
PathMNIST	0.26	49.8%	0.26	0.23	50.6%	0.27	0.22	52.8%	0.31	0.24	51.9%	0.28	0.20	54.6%	0.28	0.19	59.7%	0.33

Criteria: CIFAR-100 (MSE = 0.16, CS = 0.57); Food-101 (MSE = 0.13, CS = 0.54); PathMNIST (MSE = 0.13, CS = 0.41).

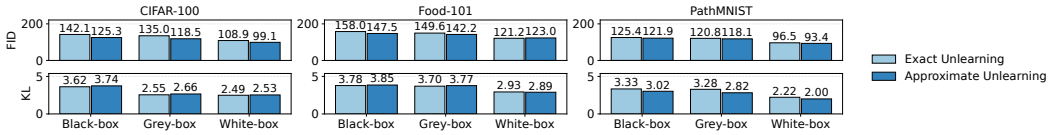


Figure 5: FID and KL scores of RETRACE from distribution-level reconstruction

73.1%). Meanwhile, CS values remain close to the criteria (e.g., Food-101: 0.49 vs. 0.54), indicating that recovered images preserve semantic similarity to the original unlearned data. These results confirm that RETRACE can reliably regenerate unlearned samples at the instance level.

At the distribution level, Figure 5 reports FID and KL divergence. Even under restrictive black-box access, RETRACE achieves reasonable alignment with the deleted distribution (e.g., CIFAR-100: FID = 142.1, KL = 3.62). Adding loss-level traces in the grey-box setting further reduces both FID and KL, and the white-box setting yields the best reconstruction quality (e.g., CIFAR-100: FID = 108.9, KL = 2.49; PathMNIST: FID = 96.5, KL = 2.22). Approximate unlearning consistently improves results, with lower FID and KL across datasets (e.g., CIFAR-100: FID = 99.1; PathMNIST: KL = 2.00), highlighting that residual traces left by approximate methods are easier to exploit. Together, these findings demonstrate that RETRACE achieves robust recovery of unlearned data across both granular instance-level signals and broader distributional structures. The detailed analyses are presented in Appendix D.3.

Comparison with baseline methods. Table 2 compares RETRACE with UIA (Hu et al., 2024) and HRec (Bertran et al., 2024) under the white-box setting, as both baselines are restricted to this access level. In the exact unlearning scenario, RETRACE achieves lower MSE (e.g., CIFAR-100: 0.20 vs. 0.35 for UIA and 0.27 for HRec), higher SR (e.g., PathMNIST: 52.8% vs. 41.0% for UIA), and higher CS (e.g., Food-101: 0.44 vs. 0.38 for UIA and 0.31 for HRec), indicating more faithful pixel- and feature-level recovery. Under approximate unlearning, RETRACE further outperforms both baselines across all metrics, with lower MSE (e.g., PathMNIST: 0.19 vs. 0.39 for UIA and 0.44 for HRec), higher SR (e.g., CIFAR-100: 73.1% vs. 59.5% and 43.0%), and the highest CS (e.g., Food-101: 0.49 vs. 0.40 and 0.33). Overall, RETRACE consistently surpasses UIA and HRec, demonstrating superior effectiveness in reconstructing unlearned data.

5.4 ABLATION STUDY

We conduct ablation studies to demonstrate the robustness and generalization of RETRACE. We also discuss its potential limitations in Appendix D.6.

Robustness. Table 3 presents the ablation results on three unlearned classes from CIFAR-100 (*Aquarium fish*, *Bed*, and *Bridge*) under the white-box setting. Across all three categories, RETRACE delivers strong performance on all metrics: MSE remains low (0.21–0.23 for exact unlearning and 0.16–0.19 for approximate unlearning), SR consistently stays above 59% and rises to around 70% in the approximate setting, and CS values remain stable in the range of 0.41–0.44. These results demonstrate that RETRACE maintains robust reconstruction ability across different unlearned data classes. Some generated cases are listed in Figure 15 in Appendix D.4.

Table 3: The effectiveness of RETRACE on different unlearned class reconstruction.

Class ID	Exact Unlearning			Approximate Unlearning		
	MSE	SR	CS	MSE	SR	CS
Aquarium fish	0.21	59.1%	0.41	0.18	70.2%	0.44
Bed	0.22	60.5%	0.42	0.19	69.2%	0.43
Bridge	0.23	60.8%	0.42	0.16	71.6%	0.42

Table 2: Effectiveness comparison of RETRACE and baselines under white-box model access.

Dataset	Exact Unlearning									Approximate Unlearning								
	UIA			HRec			**RETRACE (Ours)**			UIA			HRec			**RETRACE (Ours)**		
	MSE	SR	CS	MSE	SR	CS	MSE	SR	CS	MSE	SR	CS	MSE	SR	CS	MSE	SR	CS
CIFAR-100	0.35	46.7%	0.42	0.27	41.9%	0.31	0.20	62.4%	0.47	0.33	59.5%	0.39	0.32	43.0%	0.33	0.17	73.1%	0.50
Food-101	0.33	36.2%	0.38	0.46	27.4%	0.31	0.23	50.4%	0.44	0.31	41.4%	0.40	0.45	30.2%	0.33	0.16	65.3%	0.49
PathMNIST	0.43	41.0%	0.26	0.42	33.6%	0.22	0.22	52.8%	0.31	0.39	37.4%	0.24	0.44	37.1%	0.20	0.19	59.7%	0.33

Table 4: The effectiveness of RETRACE on text reconstruction

Unlearned Class	Black-box		Gray-box		White-box	
	BLEU	MMD	BLEU	MMD	BLEU	MMD
Sports	2.8	0.40	3.6	0.36	4.1	0.31
World	2.2	0.43	3.9	0.42	4.3	0.37
Business	2.6	0.46	3.1	0.46	4.6	0.32
Sci/Tech	2.0	0.41	3.7	0.43	4.0	0.39

Generalization. For text reconstruction, we employ a GPT-2-based (Radford et al., 2019) generator, and a DistilBERT classifier (Sanh et al., 2019) serves as the unlearned model under the exact unlearning setting. We evaluate performance on the AG News (Zhang et al., 2015) dataset using two metrics: BLEU (Papineni et al., 2002), which measures content similarity between generated and target texts, and MMD (Gretton et al., 2012), which assesses the distributional alignment across classes. As shown in Table 4, RETRACE consistently improves with stronger access levels: for example, BLEU increases from 2.8 (Sports, black-box) to 4.6 (Business, white-box), while MMD decreases across all classes (e.g., 0.46 to 0.32 for Business), indicating progressively better distributional fidelity. These results demonstrate that RETRACE effectively generalizes beyond images and is also applicable to text tasks, highlighting its broad applicability.

6 RELATED WORK

Machine unlearning. Machine unlearning (Guo et al., 2020; Bourtoule et al., 2021; Chen et al., 2022; Thudi et al., 2022; Liu et al., 2025a) enables models to forget specific training data. Existing methods are either exact or approximate: exact unlearning retrains on the retained dataset with strong guarantees but high cost (Ginart et al., 2019; Guo et al., 2020), while approximate unlearning updates parameters or gradients for efficiency (Bourtoule et al., 2021; Jia et al., 2024a; Li et al., 2024). Techniques include partition-based retraining (Bourtoule et al., 2021), gradient adjustment (Thudi et al., 2022), and adaptive methods (Gupta et al., 2021), with extensions to graphs and LLMs (Chen et al., 2022; Jia et al., 2024b; Liu et al., 2025a). However, recent work (Hu et al., 2024) shows that approximate methods can still leak forgotten knowledge, highlighting their fragility.

Reconstruction attacks. Reconstruction attacks aim to recover sensitive training data, first studied via model inversion and gradient leakage (Fredrikson et al., 2015; Zhang et al., 2023; Bertran et al., 2024; Pang et al., 2024). In unlearning, this risk is amplified, as pre–post model differences can intensify leakage (Liu et al., 2025b). Existing studies either analyze parameter shifts, which succeed in linear models but fail on deep networks, or exploit updates and gradients to optimize synthetic inputs (Ginart et al., 2019; Bourtoule et al., 2021). While recent inversion attacks show that model differentials can expose unlearned data, current approaches remain limited to instance-level recovery and do not scale to distribution-level scenarios.

7 CONCLUSION

We propose RETRACE, a reconstruction attack framework that systematically exploits residual traces left by machine unlearning. By identifying traces across access levels and leveraging them as rewards, RETRACE enables both instance- and distribution-level reconstruction. To our knowledge, this is the first demonstration of reconstruction attacks on large-scale architectures, showing effectiveness on ResNet-18 for vision and Distil-BERT for text. We provide theoretical guarantees of convergence to an exponential-tilted policy that amplifies high-trace regions, ensuring reliable recovery, and validate the approach with extensive experiments across datasets, unlearned classes, models, and unlearning methods. Our findings expose critical vulnerabilities of current unlearning techniques and highlight the need for robust unlearning mechanisms, positioning RETRACE as both a principled framework for analyzing privacy risks and a practical benchmark.

Acknowledgement of LLM Use

The use of the LLM in this work was strictly confined to linguistic refinement, including grammar and readability. All intellectual and technical contributions are the sole responsibility of the authors.

Ethics Statement

Our research adheres to the ethical principles outlined by the ICLR Code of Ethics. All experiments in this work are conducted on publicly available and fully anonymized datasets, including CIFAR-100, Food-101, PathMNIST, and AG News. These datasets are widely used in the machine learning community and do not contain personally identifiable information (PII) or sensitive attributes directly linked to real individuals. For text-based tasks, the corpus consists of news articles already curated for research purposes, where all identifying information has been removed. For medical data (i.e., PathMNIST), the dataset is part of the MedMNIST collection, which is designed to be de-identified and ethically suitable for public use. Therefore, this research does not involve the collection, processing, or exposure of human-subject data, and Institutional Review Board (IRB) approval is not required.

We further emphasize that our work focuses on methodological contributions to the study of machine unlearning and reconstruction attacks. The purpose of this research is to analyze vulnerabilities under controlled and responsible experimental settings, and to call for defenses and mitigations in subsequent studies. All experimental results are reported for the sole purpose of advancing scientific understanding of unlearning privacy issues and informing the development of more trustworthy AI systems.

Reproducibility Statement

We have taken several steps to ensure the reproducibility of our work. All datasets used in this study are publicly available benchmarks, including CIFAR-100, Food-101, PathMNIST, and AG News. The links for dataset access are provided in our paper. For theoretical analysis, all assumptions, definitions, and theorems are listed in Section 4. The complete proofs are provided in the Appendix C. The experimental settings, including unlearning techniques, model architectures, ReTrace parameter configurations, and training schedules, are fully detailed in Sections 5.1, D.1, and D.5. We report results across multiple datasets and model-access levels (black-, grey-, and white-box), and also provide ablation studies and text generalization experiments to demonstrate robustness. To facilitate replication, we release anonymized demo source code and scripts for reproducing the results presented in this work. An anonymized demo repository with our method implementation and training scripts is available at: https://anonymous.4open.science/r/ReTrace-FE5F

References

Martin Bertran, Shuai Tang, Michael Kearns, Jamie H Morgenstern, Aaron Roth, and Steven Z Wu. Reconstruction attacks on machine unlearning: Simple models are vulnerable. *Advances in Neural Information Processing Systems*, 37:104995–105016, 2024.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pp. 141–159. IEEE, 2021.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. Graph unlearning. In *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*, pp. 499–513, 2022.

Yiwei Chen, Soumyadeep Pal, Yimeng Zhang, Qing Qu, and Sijia Liu. Unlearning isn't invisible: Detecting unlearning traces in llms from model outputs. *arXiv preprint arXiv:2506.14003*, 2025.

European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council, 2016. URL https://data.europa.eu/eli/reg/2016/679/oj.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1322–1333, 2015.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Google Cloud. Safe model rollouts with canary deployments. https://cloud.google.com/blog, a. Accessed Sept 2025.

Google Cloud. Vision API release notes. https://cloud.google.com/vision/docs/release-notes, b. Accessed Sept 2025.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3832–3842, 2020.

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 34, pp. 16319–16330, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hongsheng Hu, Shuo Wang, Tian Dong, and Minhui Xue. Learn what you want to unlearn: Unlearning inversion attacks against machine unlearning. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 3257–3275. IEEE, 2024.

Jinghan Jia, Jiancheng Liu, Yihua Zhang, Parikshit Ram, Nathalie Baracaldo, and Sijia Liu. Wagle: Strategic weight attribution for effective and modular unlearning in large language models. *Advances in Neural Information Processing Systems*, 37:55620–55646, 2024a.

Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. Soul: Unlocking the power of second-order optimization for llm unlearning. In *EMNLP*, 2024b.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

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

Ziyao Liu, Huanyi Ye, Chen Chen, Yongsen Zheng, and Kwok-Yan Lam. Threats, attacks, and defenses in machine unlearning: A survey. *IEEE Open Journal of the Computer Society*, 2025b.

Meta Engineering. How Facebook ships code: Canary and shadow testing. https://engineering.fb.com/. Accessed Sept 2025.

Microsoft Azure. Cognitive services API versioning. https://learn.microsoft.com/en-us/azure/cognitive-services/cognitive-services-apis-create-account. Accessed Sept 2025.

NIST. AI risk management framework (AI RMF) pilot: Model state documentation. https://www.nist.gov/itl/ai-risk-management-framework, 2023. Accessed Sept 2025.

OpenAI. Models documentation: GPT-4 versions. https://platform.openai.com/docs/models/gpt-4. Accessed Sept 2025.

Shuchao Pang, Zhigang Lu, Haichen Wang, Peng Fu, Yongbin Zhou, and Minhui Xue. Reconstruction of differentially private text sanitization via large language models. *arXiv preprint arXiv:2410.12443*, 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Muhammad Sarmad, Hyunjoo Jenny Lee, and Young Min Kim. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5898–5907, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pp. 303–319. IEEE, 2022.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Kaiyue Zhang, Weiqi Wang, Zipei Fan, Xuan Song, and Shui Yu. Conditional matching gan guided reconstruction attack in machine unlearning. In *GLOBECOM 2023-2023 IEEE Global Communications Conference*, pp. 44–49. IEEE, 2023.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

A Threat Model

Adversary knowledge. We assume the adversary has access to both the pre-unlearning model f^+ and the post-unlearning model f^-. While this may appear strong, it is realistic in practice.

API versioning in production. Providers such as OpenAI expose multiple model versions simultaneously (e.g., `gpt-4-1106-preview` vs. `gpt-4-0125-preview`), where clients can query both before and after updates (OpenAI). Similarly, Google Cloud Vision and Microsoft Azure Cognitive Services maintain legacy API versions for backward compatibility (Google Cloud, b; Microsoft Azure). These setups allow adversaries to monitor output differences across versions, directly corresponding to access to (f^+, f^-).

Deployment practices. Large-scale platforms such as Meta and Google routinely use shadow testing and canary rollouts, where both old and new models serve traffic concurrently for days (Meta Engineering; Google Cloud, a). In regulated industries (e.g., finance, healthcare), rollback readiness is mandatory, and older checkpoints are preserved and potentially exposed through misconfiguration or insider threats (NIST, 2023).

Together, these practices show that simultaneous access to (f^+, f^-) is not only plausible but frequently realized in real-world AI deployments.

B RETRACE'S EFFICIENCY

We mathematically analyze the computational overhead of RETRACE. Its computational complexity arises from three stages. First, *trace extraction* requires evaluating discrepancies between the pre- and post-unlearning models. For each input x, computing the prediction-, loss-, and gradient-level traces involves forward and at most one backward propagation through the model, which incurs $O(C_f)$ cost, where C_f denotes the per-sample model evaluation cost. Second, in the ***RL-based generation*** stage, each policy update requires drawing N candidate samples from the generator and querying the detector for their trace scores, leading to a per-iteration complexity of $O(N \cdot C_f)$. With I RL iterations, this stage totals $O(I \cdot N \cdot C_f)$. Finally, the *reconstruction* step involves ranking and refining the N candidates, which can be done in $O(N \log N)$ time.

Overall, the end-to-end complexity of RETRACE is

$$O(I \cdot N \cdot C_f + N \log N), \tag{19}$$

dominated by the repeated generator sampling and trace evaluation. In practice, since C_f corresponds to a single model forward/backward pass, the runtime remains comparable to standard adversarial training or reinforcement-learning-based generation pipelines.

C DETAILED PROOFS

C.1 PROOF OF THEOREM 1

Proof. We consider the variational problem

$$\max_{\pi \in \Pi} \Big\{ \mathcal{J}_\tau(\pi) = \int s(x)\pi(x)\,dx - \tau \int \pi(x) \log \frac{\pi(x)}{p_0(x)}\,dx \Big\} \quad \text{s.t.} \quad \int \pi(x)\,dx = 1,\ \pi \geq 0. \tag{20}$$

Introduce a Lagrange multiplier λ and define

$$\mathcal{L}(\pi, \lambda) = \int \big[s(x)\pi(x) - \tau\, \pi(x) \log \tfrac{\pi(x)}{p_0(x)} \big]\,dx - \lambda \Big(\int \pi(x)\,dx - 1 \Big). \tag{21}$$

For any admissible direction h with $\int h = 0$, the Gâteaux derivative is

$$\frac{d}{d\epsilon} \mathcal{L}(\pi + \epsilon h, \lambda)\Big|_{\epsilon=0} = \int \Big[s(x) - \tau \big(\log \tfrac{\pi(x)}{p_0(x)} + 1 \big) - \lambda \Big]\, h(x)\,dx. \tag{22}$$

Stationarity for all h yields the Euler–Lagrange condition

$$s(x) - \tau \big(\log \tfrac{\pi^\star(x)}{p_0(x)} + 1 \big) - \lambda = 0 \implies \log \tfrac{\pi^\star(x)}{p_0(x)} = \tfrac{s(x)}{\tau} - \tfrac{\lambda}{\tau} - 1, \tag{23}$$

hence

$$\pi^\star(x) = C\, p_0(x) \exp\big(s(x)/\tau\big), \qquad C = \exp\big(-\lambda/\tau - 1\big) = \Big(\int p_0(x)\, e^{s(x)/\tau}\,dx \Big)^{-1}. \tag{24}$$

Strict concavity and uniqueness. Consider the feasible set $\mathcal{P} = \{\pi \geq 0 : \int \pi = 1\}$, which is convex. The functional $\pi \mapsto \int s\,\pi$ is linear, and the negative KL term is strictly concave:

$$\begin{aligned} \forall\, \pi_1, \pi_2 \in \mathcal{P},\ \forall\, \alpha \in (0,1): \quad & -\tau \int \big[\alpha\pi_1 + (1-\alpha)\pi_2 \big] \log \frac{\alpha\pi_1 + (1-\alpha)\pi_2}{p_0}\,dx \\ & > -\tau \Big[\alpha \int \pi_1 \log \tfrac{\pi_1}{p_0}\,dx + (1-\alpha) \int \pi_2 \log \tfrac{\pi_2}{p_0}\,dx \Big]. \end{aligned} \tag{25}$$

where the strict Jensen inequality for $\xi \log \xi$ applies unless $\pi_1 = \pi_2$ a.e. Therefore $\mathcal{J}_\tau$ is strictly concave over $\mathcal{P}$, implying the stationary solution $\pi^\star$ is the unique global maximizer in Π. This completes the proof. □

C.2 Proof of Theorem 2

Proof. Fix $\varepsilon > 0$ and define the ε-neighborhood of the deleted support

$$A_\varepsilon := \mathcal{N}_\varepsilon\big(\mathrm{supp}(\mathbb{P}_{\mathrm{del}})\big) = \big\{x \in \mathcal{X} : \exists\, x_0 \in \mathrm{supp}(\mathbb{P}_{\mathrm{del}}) \text{ s.t. } d_{\mathcal{X}}(x, x_0) \le \varepsilon\big\}. \tag{26}$$

Because $\mathbb{P}_{\mathrm{del}}(A_\varepsilon) > 0$ for any $\varepsilon > 0$ and $\mathbb{P}_{\mathrm{del}} \ll p_0$ by Assumption 1, we have $p_0(A_\varepsilon) > 0$. From Theorem 1, $\pi^\star(x) = Z_\tau^{-1} p_0(x) \exp(s(x)/\tau)$ with $Z_\tau = \int p_0 e^{s/\tau}$, hence

$$p_\star := \pi^\star(A_\varepsilon) = \frac{\int_{A_\varepsilon} p_0(x) e^{s(x)/\tau}\, dx}{\int_{\mathcal{X}} p_0(x) e^{s(x)/\tau}\, dx} \ge \frac{e^{\inf_{x\in A_\varepsilon} s(x)/\tau}\, p_0(A_\varepsilon)}{e^{\sup_{x\in\mathcal{X}} s(x)/\tau}} > 0, \tag{27}$$

since $s \in [0,1]$ and $p_0(A_\varepsilon) > 0$. Let $X_1, \dots, X_k$ be i.i.d. draws from $\pi^\star$. The event $\{X_j \notin A_\varepsilon,\ \forall j\}$ has probability $(1 - p_\star)^k$. Therefore

$$\Pr\Big[\exists j \le k :\ X_j \in A_\varepsilon\Big] = 1 - (1 - p_\star)^k. \tag{28}$$

By definition of A_ε, for any such X_j there exists $x \in \mathcal{D}_{\mathrm{del}}$ with $d_{\mathcal{X}}(X_j, x) \le \varepsilon$. Let the refinement operator in §3.3 be denoted $\mathcal{R}$; assume it is L-Lipschitz and uses stepsize $\eta > 0$ that respects the neighborhood, i.e.,

$$\|\mathcal{R}(X_j) - X_j\|_{\mathcal{X}} \le \eta \quad \text{and} \quad \eta \le \varepsilon - d_{\mathcal{X}}(X_j, \mathcal{D}_{\mathrm{del}}). \tag{29}$$

Then $\hat{X}_j := \mathcal{R}(X_j) \in A_\varepsilon$ and $\min_{x\in\mathcal{D}_{\mathrm{del}}} d_{\mathcal{X}}(\hat{X}_j, x) \le \varepsilon$. Hence

$$\Pr\Big[\exists j \le k :\ \min_{x\in\mathcal{D}_{\mathrm{del}}} d_{\mathcal{X}}(\hat{X}_j, x) \le \varepsilon\Big] \ \ge\ 1 - (1 - p_\star)^k. \tag{30}$$

Finally, to attain target confidence $1 - \delta$, it suffices to choose $k \ge \log(1/\delta)/p_\star$. This proves the claim. □

C.3 Proof of Theorem 3

Proof. Let $\pi^\star$ be the unique maximizer given by Theorem 1. Let $\widehat{P}_k := \frac{1}{k}\sum_{j=1}^k \delta_{\hat{X}_j}$ be the empirical distribution of the k reconstructed samples after the local refinement $\mathcal{R}$ in §3.3.

Step 1: Bias–variance decomposition for a general IPM.. Let $d(\cdot,\cdot)$ be any integral probability metric (IPM) induced by a function class $\mathcal{F}$:

$$d(P, Q) := \sup_{f\in\mathcal{F}} \Big|\mathbb{E}_P[f] - \mathbb{E}_Q[f]\Big|. \tag{31}$$

By the triangle inequality,

$$d(\widehat{P}_k, \mathbb{P}_{\mathrm{del}}) \le d(\widehat{P}_k, \pi^\star) + d(\pi^\star, \mathbb{P}_{\mathrm{del}}). \tag{32}$$

Moreover, if the refinement $\mathcal{R}$ is L-Lipschitz and maps pre-selected candidates $\widetilde{X}_j$ to $\hat{X}_j = \mathcal{R}(\widetilde{X}_j)$, then standard stability for IPMs on bounded domains implies (for a Lipchitz-bounded function class $\mathcal{F}$)

$$d(\widehat{P}_k, \pi^\star) \ \le\ C_L\, d(\widetilde{P}_k, \pi^\star), \qquad \widetilde{P}_k := \frac{1}{k}\sum_{j=1}^k \delta_{\widetilde{X}_j}, \tag{33}$$

for some constant $C_L \ge 1$ depending only on L and the diameter of $\mathcal{X}$. Thus it suffices to control $d(\widetilde{P}_k, \pi^\star)$ and $d(\pi^\star, \mathbb{P}_{\mathrm{del}})$.

Step 2: Sampling (variance) term $d(\widetilde{P}_k, \pi^\star)$**.**. For common choices of d, we have non-asymptotic concentration with explicit rates:

- **MMD.** If $\mathcal{F}$ is the unit ball of an RKHS with bounded kernel k and $\sup_x k(x,x) \le K$, then by standard RKHS concentration (e.g., via McDiarmid), for any $\delta \in (0,1)$,

$$\Pr\left[\mathrm{MMD}(\widetilde{P}_k, \pi^\star) \le 2\sqrt{\frac{K\log(2/\delta)}{k}}\right] \ge 1 - \delta. \tag{34}$$

- W_1. If $\mathcal{X} \subset \mathbb{R}^d$ is bounded with diameter D, the Fournier–Guillin bound gives constants $C_d > 0$ such that, for any $\delta \in (0, 1)$,

$$\Pr\left[W_1(\widetilde{P}_k, \pi^\star) \le C_d \left(\frac{\log(2/\delta)}{k}\right)^{1/d}\right] \ge 1 - \delta. \tag{35}$$

For a general IPM with functions bounded by $\|f\|_\infty \le B$, symmetrization plus Hoeffding inequality yields the generic bound

$$\Pr\left[d(\widetilde{P}_k, \pi^\star) \le 2\mathfrak{R}_k(\mathcal{F}) + B\sqrt{\frac{2\log(2/\delta)}{k}}\right] \ge 1 - \delta, \tag{36}$$

where $\mathfrak{R}_k(\mathcal{F})$ is the empirical Rademacher complexity (often $O(1/\sqrt{k})$).

Step 3: Bias term $d(\pi^\star, \mathbb{P}_{\text{del}})$ via exponential tilting.. From Theorem 1, $\pi^\star$ has density

$$\pi^\star(x) = \frac{p_0(x)\exp(s(x)/\tau)}{Z_\tau}, \qquad Z_\tau := \int p_0(x)\exp(s(x)/\tau)\,dx. \tag{37}$$

We first derive an upper bound for $d_{\text{TV}}(\pi^\star, \mathbb{P}_{\text{del}})$ via an upper bound on $\text{KL}(\mathbb{P}_{\text{del}}\|\pi^\star)$ and then apply Pinsker. By the chain rule for KL,

$$\text{KL}(\mathbb{P}_{\text{del}}\|\pi^\star) = \text{KL}(\mathbb{P}_{\text{del}}\|p_0) - \frac{1}{\tau}\,\mathbb{E}_{\mathbb{P}_{\text{del}}}[s] + \log Z_\tau. \tag{38}$$

Since $s \in [0, 1]$, Hoeffding's lemma gives (for any base measure)

$$\log Z_\tau = \log \mathbb{E}_{p_0}\left[\exp\left(\tfrac{s}{\tau}\right)\right] \le \frac{\mathbb{E}_{p_0}[s]}{\tau} + \frac{1}{8\tau^2}. \tag{39}$$

Plugging equation 39 into equation 38 yields the explicit upper bound

$$\text{KL}(\mathbb{P}_{\text{del}}\|\pi^\star) \le \text{KL}(\mathbb{P}_{\text{del}}\|p_0) + \frac{1}{\tau}\left(\mathbb{E}_{p_0}[s] - \mathbb{E}_{\mathbb{P}_{\text{del}}}[s]\right) + \frac{1}{8\tau^2}. \tag{40}$$

By Pinsker's inequality,

$$d_{\text{TV}}(\pi^\star, \mathbb{P}_{\text{del}}) \;\le\; \sqrt{\tfrac{1}{2}\,\text{KL}(\mathbb{P}_{\text{del}}\|\pi^\star)}. \tag{41}$$

Therefore

$$d_{\text{TV}}(\pi^\star, \mathbb{P}_{\text{del}}) \;\le\; \sqrt{\frac{1}{2}\left(\text{KL}(\mathbb{P}_{\text{del}}\|p_0) + \frac{\mathbb{E}_{p_0}[s] - \mathbb{E}_{\mathbb{P}_{\text{del}}}[s]}{\tau} + \frac{1}{8\tau^2}\right)}. \tag{42}$$

Relating separability to the bias. By Definition 2 (trace separability), $\mathbb{E}_{\mathbb{P}_{\text{del}}}[s] \ge \mathbb{E}_{\mathbb{P}_{\text{ret}}}[s] + \Delta$. If p_0 is a public prior independent of the unlearning operation, we can treat $\mathbb{E}_{p_0}[s]$ as a constant that does not increase with the separability margin; hence the difference $\mathbb{E}_{p_0}[s] - \mathbb{E}_{\mathbb{P}_{\text{del}}}[s]$ decreases as Δ grows, which tightens equation 42. Consequently, the total variation bias can be controlled by (τ, Δ) and $\text{KL}(\mathbb{P}_{\text{del}}\|p_0)$.

Step 4: From TV to the chosen metric $d(\cdot, \cdot)$.. On a bounded domain $\mathcal{X}$ with diameter D,

$$W_1(\pi^\star, \mathbb{P}_{\text{del}}) \;\le\; D \cdot d_{\text{TV}}(\pi^\star, \mathbb{P}_{\text{del}}). \tag{43}$$

For MMD with kernel k bounded by K,

$$\begin{aligned}\text{MMD}(\pi^\star, \mathbb{P}_{\text{del}}) &= \sup_{\|f\|_{\mathcal{H}} \le 1} \left|\mathbb{E}_{\pi^\star}[f] - \mathbb{E}_{\mathbb{P}_{\text{del}}}[f]\right| \\ &\le \sup_{\|f\|_{\mathcal{H}} \le 1} \|f\|_\infty \cdot 2\, d_{\text{TV}}(\pi^\star, \mathbb{P}_{\text{del}}) \\ &\le 2\sqrt{K}\; d_{\text{TV}}(\pi^\star, \mathbb{P}_{\text{del}}).\end{aligned} \tag{44}$$

using $\|f\|_\infty \le \sqrt{K}\|f\|_{\mathcal{H}}$. More generally, for any IPM induced by a function class $\mathcal{F}$ with $\|f\|_\infty \le B$,

$$d(\pi^\star, \mathbb{P}_{\text{del}}) \;\le\; 2B\; d_{\text{TV}}(\pi^\star, \mathbb{P}_{\text{del}}). \tag{45}$$

Combine with equation 42 to obtain an explicit bias bound:

$$d(\pi^\star, \mathbb{P}_{\text{del}}) \le C_{\text{met}} \cdot \sqrt{\tfrac{1}{2}\left(\mathrm{KL}(\mathbb{P}_{\text{del}} \| p_0) + \tfrac{\mathbb{E}_{p_0}[s] - \mathbb{E}_{\mathbb{P}_{\text{del}}}[s]}{\tau} + \tfrac{1}{8\tau^2}\right)}, \tag{46}$$

$$\text{where } C_{\text{met}} = \begin{cases} D, & d = W_1, \\ 2\sqrt{K}, & d = \text{MMD}, \\ 2B, & \text{general } d. \end{cases}$$

Define $C_1(\tau, \Delta)$ as the right-hand side; it decreases as $\tau \downarrow 0$ and as the separability margin Δ increases (since $\mathbb{E}_{\mathbb{P}_{\text{del}}}[s]$ increases with Δ).

Step 5: Put together.. With probability at least $1 - \delta$, we have

$$d(\widetilde{P}_k, \pi^\star) \;\le\; \epsilon(k, \delta), \tag{47}$$

where $\epsilon(k, \delta)$ is the sampling error of Step 2 (e.g., $2\sqrt{K \log(2/\delta)/k}$ for MMD, or $C_d(\log(2/\delta)/k)^{1/d}$ for W_1). Hence, by equation 32 and the stability of $\mathcal{R}$,

$$d(\widehat{P}_k, \mathbb{P}_{\text{del}}) \;\le\; C_L\, \epsilon(k, \delta) \;+\; C_1(\tau, \Delta), \tag{48}$$

which proves the theorem. □

D Evaluation

D.1 Detailed Experimental Setup

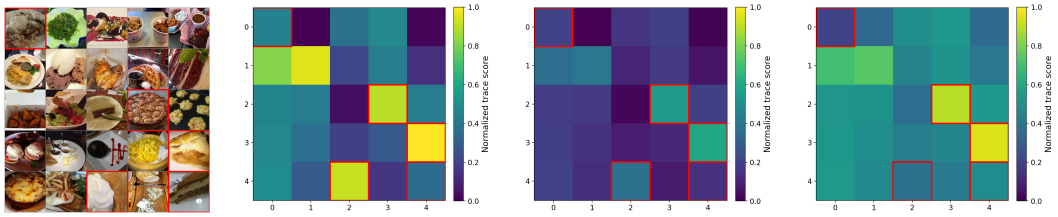
Unlearning method. Following existing work (Hu et al., 2024), to obtain the paired models (f^+, f^-), we adopt two categories of unlearning procedures. *Exact unlearning* is implemented by removing the forgotten samples and fine-tuning the model on the remaining data for the same number of epochs as the original training. *Approximate unlearning* is implemented using the single gradient unlearning method (Thudi et al., 2022). In both cases, unlearning is performed in a *class-wise* manner, which reflects realistic scenarios where requests often target all samples belonging to a specific semantic category. We set class 0 as the default target for unlearning.

Datasets. We evaluate on CIFAR-100 with 50k training images and 10k test images of size 32×32 RGB. This dataset contains 100 object categories with relatively low resolution and high intra-class variability, making it a standard benchmark for image classification. Food-101 includes 75k training images and 25k test images of size 224×224 RGB. It covers 101 food categories collected from real-world scenarios, characterized by large intra-class diversity, occlusion, and noisy labels. PathMNIST, a subset of the MedMNIST collection, consists of 89,996 training images, 7,180 validation images, and 7,180 test images of size 28×28 grayscale. It provides 9 classes derived from colorectal cancer histology slides, capturing diverse tissue types such as adipose, lymphocytes, smooth muscle, and adenocarcinoma epithelium.

Models (f^+, f^-). We adopt ResNet-18 on CIFAR-100, Food-101, and PathMNIST datasets for image classification tasks. We use cross-entropy loss with the SGD optimizer and a cosine learning rate schedule, and maintain a checkpoint pair (f^+, f^-) for each dataset.

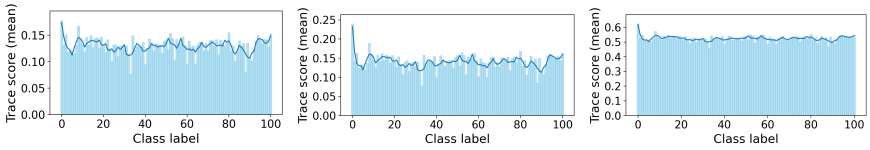
Generator (RL Policy π_ϕ). We use a DCGAN-style generator G that maps a latent vector $z \in \mathbb{R}^d$ to 32×32 images with $\tanh$ outputs in $[-1, 1]$, and an RL policy π_ϕ that produces z via a diagonal Gaussian whose mean and log-std are predicted by a two-layer MLP. At each step, the policy samples $z = \mu_\phi(\epsilon) + \exp(\log \sigma_\phi(\epsilon)) \odot \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$, generates $x_{\text{fake}} = G(z)$, and receives a reward built from unlearning traces between f^+ and f^-. The policy is optimized with PPO (clipped objective with entropy bonus), while G can be jointly fine-tuned by a surrogate loss that maximizes the trace signal and includes teacher-guided KL distillation from f^+ and f^-, and total-variation regularization. In practice, we initialize G with different pretrained GANs depending on the dataset: an ImageNet-pretrained GAN for CIFAR-100 and Food-101, and a MedMNIST-pretrained GAN for PathMNIST.

ReTrace settings. We set $\alpha = 0.4$, $\beta = 0.2$, and $\gamma = 0.4$ as the relative weights for trace components. For reconstruction, we select samples using top-k with $k = 64$.



(a) Original Data (b) Black-box (c) Grey-box (d) White-box

Figure 6: Traces of unlearned data under different model access levels on Food-101 in the approximate unlearning scenario. The images in red boxes represent the unlearned data "*Apple pie*".



(a) Black-box (b) Grey-box (c) White-box

Figure 7: Trace distribution under different model access levels on Food-101 in the exact unlearning scenario. Class 0 represents "*Apple pie*".

Evaluation metrics. For instance-level evaluation, we use *Mean Squared Error (MSE)* to assess the *pixel-level* difference between two images; *cosine similarity (CS)* to evaluate the *feature-level* difference; *success rate (SR)* to evaluate the percentage of generated images being unlearned class. For distribution-level evaluation, we use *Fréchet Inception Distance (FID)* to evaluate the visual and *statistical similarity* between reconstructed and original distributions, and *Kullback–Leibler (KL) divergence* to assess the alignment of their *probability distributions*.

Baseline methods. We compare RETRACE with two state-of-the-art reconstruction attack methods, which are listed as follows.

- **Unlearning Inversion Attack (UIA)** (Hu et al., 2024), which is conducted to recover unlearned data in white-box access.
- **HRec** (Bertran et al., 2024), which achieves nearly-perfect attack on linear regression and can be generalized to other model architectures.

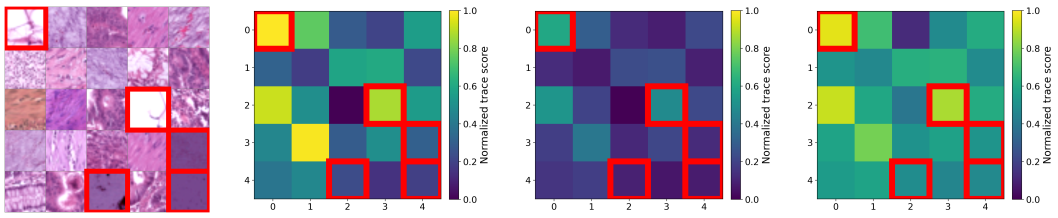
D.2 TRACE VISUALIZATION

For both Food-101 and PathMNIST, we present the $f^+ - f^-$ trace heatmaps and the corresponding trace distributions (Figures 6, 7, 8, 9).

D.3 EFFECTIVENESS OF RETRACE

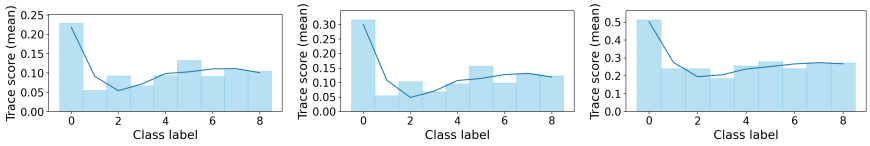
Instance level. Table 1 presents the instance-level reconstruction results of RETRACE across three datasets and three model-access levels. For the criteria of MSE and CS, we compute all pairwise distances within the unlearned class in the original dataset and take the average, which reflects the natural intra-class variability (values are presented in tablenote).

Under the *exact unlearning* setting, across three datasets, the reconstructed samples achieve MSE values that are only slightly higher than the intra-class criteria (e.g., CIFAR-100: 0.20 vs. 0.16; Food-101: 0.23 vs. 0.13), showing that the pixel-level differences between reconstructions and unlearned images remain small. The SR values are consistently around 50–60%, indicating that our method can successfully generate a large proportion of samples that are classified as belonging to the unlearned class. Meanwhile, the CS values (e.g., CIFAR-100: 0.47 vs. 0.57) demonstrate that the reconstructed samples lie close to the original unlearned data in feature space, confirming that the recovered images retain strong semantic similarity.



(a) Original Data (b) Trace under Black-box (c) Trace under Grey-box (d) Trace under White-box

Figure 8: Traces of unlearned data under different model access levels on PathMNIST in the approximate unlearning scenario. The images in red boxes represent the unlearned data "*Adipose*".



(a) Black-box (b) Grey-box (c) White-box

Figure 9: Trace distribution under different model access levels on PathMNIST in the exact unlearning scenario. Class 0 represents "*Adipose*".

In the *approximate unlearning* scenario, the performance further improves across datasets and access levels. MSE decreases noticeably compared to exact unlearning (e.g., CIFAR-100: 0.17 vs. 0.20), suggesting that approximate unlearning leaves stronger residual signals for pixel-level recovery. SR also increases significantly, with white-box access achieving the highest rates (e.g., CIFAR-100: 73.1%; PathMNIST: 59.7%), demonstrating that our method can generate an even larger number of valid samples for the forgotten class. Finally, CS values are consistently close to the criteria (e.g., Food-101: 0.49 vs. 0.54), indicating that reconstructed images not only recover visual details but also align well with the semantic representations of unlearned data.

Distribution level. Figure 5 illustrates the distribution-level reconstruction results of ReTrace in terms of FID and KL divergence.

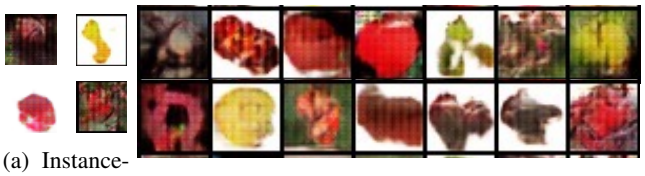
Under exact unlearning, across all three datasets, ReTrace achieves meaningful reconstruction performance under all access levels. In the black-box setting, both FID and KL remain at reasonably low values (e.g., CIFAR-100: FID = 142.1, KL = 3.62; PathMNIST: FID = 125.4, KL = 3.33), showing that the method can approximate the deleted distribution even with limited information. Moving to the grey-box setting, the incorporation of loss-level traces consistently reduces both FID and KL (e.g., CIFAR-100: FID drops from 142.1 to 135.0, KL from 3.62 to 2.55), demonstrating improved alignment with the original distribution. The best results are obtained in the white-box setting, where gradient-level traces further enhance reconstruction quality, yielding the lowest FID and KL across datasets (e.g., CIFAR-100: FID = 108.9, KL = 2.49; PathMNIST: FID = 96.5, KL = 2.22). These results highlight that while ReTrace is effective even under restrictive black-box conditions, more informative access significantly boosts distribution-level recovery.

In the approximate unlearning setting, the results further improved. Compared to exact unlearning, FID values are consistently lower (e.g., CIFAR-100: 99.1 vs. 108.9; PathMNIST: 93.4 vs. 96.5), reflecting that approximate unlearning leaves stronger distributional traces that ReTrace can exploit. KL divergence also decreases slightly in most cases (e.g., PathMNIST: 2.00 vs. 2.22), confirming that the reconstructed samples align closely with the underlying class distribution.

Cases. We present reconstructed samples at the *instance level* for CIFAR-100, Food-101, and PathMNIST under the white-box setting.

For CIFAR-100, we present reconstructed samples under *Exact Unlearning* (Figure 10).

For Food-101, we present reconstructed samples under *Exact Unlearning* (Figure 12) and under *Approximate Unlearning* (Figure 11).



(a) Instance-level. (b) Distribution-level.

Figure 10: Generated unlearned samples ("Apple class") in CIFAR-100 under exact unlearning.

(a) Instance-level. (b) Distribution-level.

Figure 11: Generated unlearned samples ("Apple pie class") in Food-101 under approximate unlearning.

For PathMNIST, we present reconstructed samples under *Exact Unlearning* (Figure 14) and under *Approximate Unlearning* (Figure 13).

D.4 Ablation on Unlearned Classes

We present a group of examples on *reconstructed samples* for multiple CIFAR-100 classes under *Exact* and *Approximate* unlearning, as shown in Figure 15.

D.5 Generalization on Text Dataset

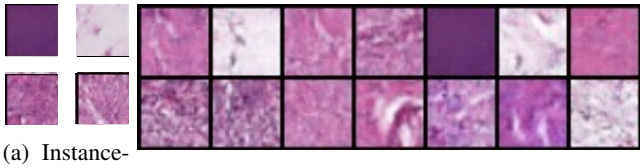
Setup. We employ a pretrained GPT-2 (small) decoder as the text generator and fine-tune it with a lightweight PPO/REINFORCE objective, while the DistilBERT classifiers (Sanh et al., 2019) serve as unlearned model: f^+ trained on the full AG News corpus and f^- retrained after *Exact Unlearning* of the unlearned class. At reconstruction time, we query f^+ and f^- on sampled texts to construct a *trace* reward whose components depend on the access setting: Black-box uses the prediction divergence $\delta_{\text{pred}}(x) = \|\text{softmax}(f^+(x)) - \text{softmax}(f^-(x))\|_2$; Gray-box augments this with a loss gap $\delta_{\text{loss}}(x) = \left|\ell(f^+(x), \hat{y}) - \ell(f^-(x), \hat{y})\right|$ where $\hat{y} = \arg\max f^+(x)$; White-box additionally incorporates a representation discrepancy $\delta_{\text{feat}}(x) = 1 - \cos\big(h^+(x), h^-(x)\big)$ from hidden states. The total reward is the weighted sum of the trace term, a class-prior from f^+ (target-class probability), a discriminative term that increases the target-class logit, a small fluency bonus from the policy log-likelihood, and a length penalty. We decode with top-k sampling, a repetition penalty, and a minimum new-token budget to mitigate collapse. We evaluate reconstruction effectiveness with BLEU, computed as corpus-level n-gram overlap against held-out texts from the forgotten class, and MMD, computed between generated and forgotten-class distributions in the DistilBERT `[CLS]` embedding space using a multi-bandwidth RBF kernel.

Reconstruction examples on the text dataset. The representative reconstructions on AG News (forgotten class: *Sports*) are shown below. These examples illustrate the recovered sports style.

Sample 0: Inter lift Coppa Italia after 1–0 final; Martínez scores decisive header.
Sample 1: Celtics survive Heat 104–99; Tatum posts 34–9–7 in Game 5.
Sample 2: France edge Spain 1–0 to lift UEFA Nations League crown in Milan.

(a) Instance-level. (b) Distribution-level.

Figure 12: Generated unlearned samples ("Apple pie class") in Food-101 under exact unlearning.



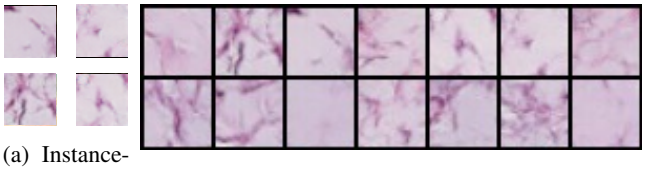
(a) Instance-level. (b) Distribution-level.

Figure 13: Generated unlearned samples ("Adipose class") in PathMNIST under approximate unlearning.

D.6 Discussion of Potential Limitations

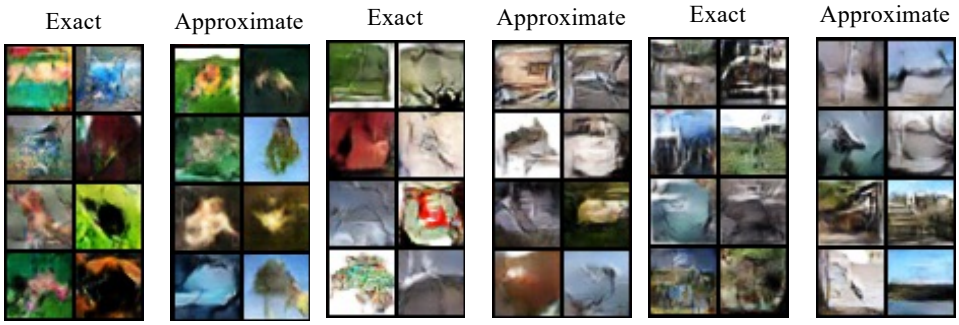
Our method builds upon pretrained generative models, which serve as the base model for reconstructing the unlearned content. This design alleviates the challenges of training from scratch, such as unstable optimization and mode collapse, and enables more efficient adaptation with RL signals. Nevertheless, the characteristics of the pretrained model itself naturally influence the reconstruction quality. In particular, the alignment between the pretraining corpus and the target task domain plays a critical role: if the pretraining data diverges significantly from the forgotten distribution, the generated samples may deviate from the intended semantics or result in unsatisfactory outputs.

However, in modern machine learning practice, the use of pretrained models has become standard. Large-scale pretraining not only reduces computational overhead but also greatly improves practicality compared to training from scratch. Therefore, this potential limitation, which could otherwise affect ReTrace's performance, is unlikely to pose a significant concern.



(a) Instance-level. (b) Distribution-level.

Figure 14: Generated unlearned samples (“Adipose class”) in PathMNIST under exact unlearning.



(a) Class 1: Aquarium fish. (b) Class 5: Bed. (c) Class 12: Bridge.

Figure 15: Reconstructed unlearned samples on class 1, 5 and 12 on CIFAR-100.