URGE: A UNIFIED REPRESENTATION-GUIDED LOSS REWEIGHTING FRAMEWORK FOR FINE-GRAINED MACHINE UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning is an emerging paradigm to remove specified data influences from trained models, with broad applications across image classification, image generation and text generation. A prominent direction is gradient-based unlearning (GU), which employs forgetting objectives and retain regularization to balance forgetting and utility. Loss reweighting further improves GU by explicitly assigning different update strengths to samples or tokens based on their forgetting needs. However, existing methods face two key limitations: (i) effectiveness: prediction-distribution-based methods rely on signals that are partially redundant with the original optimization gradients; (ii) applicability: other methods require external auxiliary models, retain data, or task-specific designs. In response, we propose **URGE**, a **U**nified **R**epresentation-**G**uided r**E**weighting framework for GU. URGE constructs weights for each sample/token from two representation-based signals: Conditional Representation Invariance estimates its intrinsic need for forgetting, while Representation Similarity estimates its residual need for further forgetting. By leveraging internal representations, URGE captures richer information about how samples/tokens are encoded in the model, improving unlearning effectiveness. Without requiring external models or retain data, URGE is applicable across GU methods and tasks. Experiments against 18 baselines across 3 tasks and 13 model-dataset settings show that URGE generally achieves a better forgetting-utility trade-off than GU and reweighting baselines.

1 INTRODUCTION

Machine unlearning (MU) enables efficient removal of specific targets from trained models (Bourtoule et al., 2021), including training samples, classes, concepts or undesired behaviors across image classification, image generation, and text generation. MU emerged in response to the General Data Protection Regulation (GDPR) (Voigt & Von dem Bussche, 2017), which grants users the right to withdraw personal data, thus reducing the risk of privacy leakage. MU also serves security and reliability purposes by unlearning poisoned, copyrighted or harmful data. MU shares a common goal: striking a forgetting–utility trade-off—making the model behave as if the target had never been learned, while preserving utility on retain knowledge.

MU is classified as exact or approximate based on whether it guarantees strict indistinguishability from the retrain oracle. Due to the high cost of exact MU, research primarily focuses on approximate MU, with gradient-based unlearning (GU) being a prominent paradigm. GU improves the forgetting-utility trade-off through tailored forgetting losses or retain regularization, with implementations across tasks, including image classification (Chen et al., 2023; Chundawat et al., 2023; Kurmanji et al., 2023), image generation (Zhang et al., 2024a; Gandikota et al., 2023; Wu et al., 2025), and text generation (Yao et al., 2024; Zhang et al., 2024b; Fan et al., 2026; Li et al., 2024).

To further achieve fine-grained unlearning, recent studies propose loss reweighting to control the forgetting strength of individual samples or tokens. Some methods filter forget samples—essentially assigning binary $\{0, 1\}$ weights per sample—to improve unlearning effectiveness or efficiency, e.g., FUNU (Li et al., 2025b), UPCORE (Patil et al., 2025), and LLM-Coreset (Pal et al., 2025). Others assign continuous weights to individual sample/token losses, e.g., WGA (Wang et al., 2025c),

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

SatImp (Yang et al., 2025), and LoReUn (Li et al., 2025a). The core principle of loss reweighting is to allocate forgetting budget on demand, thereby improving the forgetting–utility Pareto frontier.

Existing reweighting methods can be broadly grouped by the signals used to construct weights. (i) Prediction-distribution-based methods derive weights from the model's output on forget data, such as predictive probability (Yu et al., 2026; Tran et al., 2025; Wu & Zhou, 2026; Yang et al., 2026), loss (Wang et al., 2025d; Liu et al., 2025), or entropy (Koh et al., 2026; Zhai et al., 2026). Although simple and efficient, these methods face an effectiveness limitation caused by signal redundancy: their weights essentially reuse predictive signals already encoded in the forgetting objective and gradients, providing limited information beyond the original optimization. (ii) Semantic-selection-based methods (Wan et al., 2025; Wang et al., 2025a; Zhou et al., 2026) rely on external models to identify critical samples/tokens, incurring extra costs that limit their applicability. (iii) Forget-retain-conflict-based methods (Zhao et al., 2025; Niu et al., 2025; Yüce et al., 2026) directly measure the influence of samples/tokens on forgetting and retention, but the need for retain data limits their applicability. Furthermore, existing studies mostly focus on specific tasks, leaving the general applicability of loss reweighting underexplored.

In this paper, we propose **URGE**, a **U**nified **R**epresentation **G**uided r**E**weighting framework for gradient-based unlearning. Requiring no external models or retain data, it is applicable to diverse GU methods and tasks. By deriving weights from internal representations, URGE leverages high-dimensional information about how each forgetting sample/token is encoded inside the model, mitigating signal redundancy, thus improving effectiveness. Specifically, we introduce a static Conditional Representation Invariance (CRI) score to capture the intrinsic forgetting need of each sample/token—that is, how strongly it is memorized and associated with the target knowledge. Meanwhile, we propose a dynamic Representation Similarity (RS) score to reflect the residual forgetting need of each sample/token—that is, how strongly it is still memorized by the model at the current unlearning step. Combining both scores, URGE assigns larger weights to samples/tokens with high intrinsic and residual forgetting needs, and adaptively reduces their weights as the residual need declines, eventually reaching an early stop, thus enabling fine-grained control over the forgetting–utility trade-off. Our contributions are summarized as follows.

- We propose a representation-guided loss reweighting method for unlearning, which derives forgetting weights from internal representations instead of output distributions, requiring neither retain data nor external models.
- We formulate loss reweighting for gradient-based unlearning from a holistic view, establishing a unified framework across tasks and GU methods.
- We validate URGE via experiments against 18 baselines across 3 tasks with total 13 model-dataset settings, showing its broad applicability and stronger performance than existing reweighting methods and GU baselines.

2 Preliminaries: A Unified Formulation of Machine Unlearning

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be a training dataset, where (x_i, y_i) is a training example. The forget set is $\mathcal{D}_f \subset \mathcal{D}$, and the retain set is $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$. Let θ_{ori} be the original model trained on $\mathcal{D}$. An unlearning algorithm $\mathcal{U}$ produces an unlearned model θ_{unl} which should approximate a model θ_{retr} retrained from scratch using only $\mathcal{D}_r$. MU aims to remove the knowledge of $\mathcal{D}_f$ while preserving utility on $\mathcal{D}_r$. Formally, GU updates the model by minimizing an overall unlearning objective:

$$\mathcal{L}_{\text{MU}}(\theta) = \mathcal{L}_{\text{forget}}(\theta; \mathcal{D}_f) + \lambda \mathcal{L}_{\text{retain}}(\theta; \mathcal{D}_r), \tag{1}$$

where $\mathcal{L}_{\text{forget}}$ is the forgetting loss, $\mathcal{L}_{\text{retain}}$ is usually the original task loss on $\mathcal{D}_r$, and λ controls the balance. Under this unified framework, tasks are characterized by the definitions of $\mathcal{D}_f$ and $\mathcal{L}_{\text{forget}}$.

MU for Image Classification. Here, each example (x_i, y_i) consists of an input image x_i and a class label y_i. The forget set $\mathcal{D}_f$ is constructed from target-class samples or randomly selected samples. As the classifier f_θ is typically trained with cross-entropy loss, a common forgetting loss is gradient ascent on this loss over $\mathcal{D}_f$: $\mathcal{L}_{\text{GA}}^{\text{cls}}(\theta; \mathcal{D}_f) = -\mathcal{L}_{\text{CE}}(\theta; \mathcal{D}_f)$.

MU for Image Generation in Diffusion Models (DMs). In DMs, each example (x_i, y_i) consists of an image x_i and a conditioning signal y_i. The model learns a denoising network $\epsilon_\theta(x_t, t, y_i)$ to

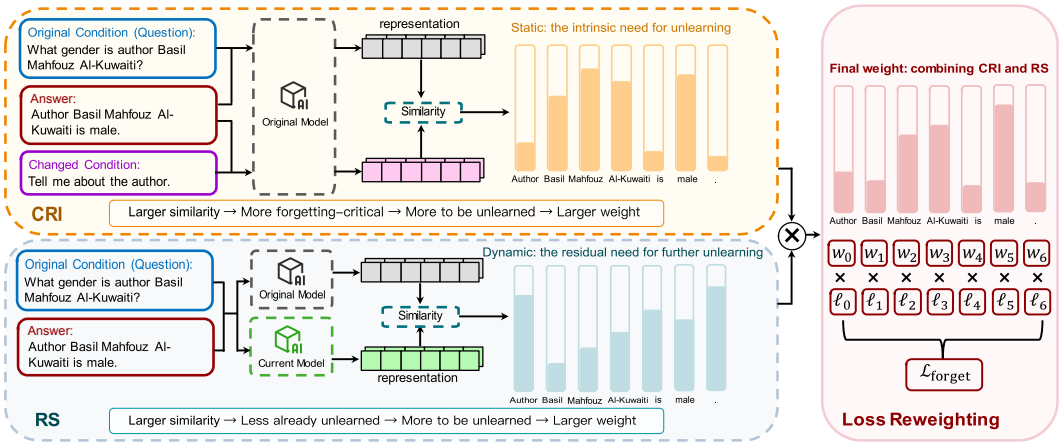


Figure 1: Overview. Taking LLM unlearning as an example, URGE derives token-level forgetting weights from two scores. The upper branch computes CRI by comparing representations under the original and perturbed conditions, where a larger bar indicates higher CRI and a stronger intrinsic need for forgetting. The lower branch computes RS by comparing the current and pre-unlearning representations, where a larger bar indicates less-forgotten tokens that require further forgetting.

predict the noise added at diffusion step t. The forward noising process is $x_t = \alpha_t x_0 + \sigma_t \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$, α_t and σ_t are determined by the noise schedule.

In this task, $\mathcal{D}_f$ is specified by a target condition set $\mathcal{Y}_f$, where each $y_f \in \mathcal{Y}_f$ is a target concept. For a class-conditioned DDPM, y_f is a class label; for a text-conditioned Stable Diffusion model, y_f can be a class-name prompt. Thus, the forget set is $\mathcal{D}_f = \{(x_i, y_i) \in \mathcal{D} \mid y_i \in \mathcal{Y}_f\}$. A basic forgetting loss is random labeling (RL). Let $\mathcal{Y}_r$ denote the set of non-target conditions. For each (x_i, y_i), we replace c_i with a $\tilde{y}_i \sim \text{Unif}(\mathcal{Y}_r)$, forming $\widetilde{\mathcal{D}}_f = \{(x_i, \tilde{y}_i)\}_{i=1}^N$. Based on the standard diffusion noise-prediction loss, the random labeling loss is $\mathcal{L}_{\text{RL}}^{\text{DM}}(\theta; \mathcal{D}_f) = \mathbb{E}_{(x_0,\tilde{y})\sim\widetilde{\mathcal{D}}_f, t, \epsilon} \|\epsilon - \epsilon_\theta(x_t, t, \tilde{y})\|_2^2$.

MU for Text Generation in Large Language Models (LLMs). The forget set $\mathcal{D}_f$ consists of target text passages or question–answer pairs. We write both cases uniformly as (x_i, y_i), where x_i can be empty for standalone passages, and $y_i = (y_i^1, \ldots, y_i^{|y_i|})$ denotes the target output tokens.

As LLMs are commonly trained with the negative log-likelihood objective, a common forgetting loss is gradient ascent on this loss over $\mathcal{D}_f$: $\mathcal{L}_{\text{GA}}^{\text{LLM}}(\theta; \mathcal{D}_f) = \frac{1}{|\mathcal{D}_f|} \sum_{(x,y)\in\mathcal{D}_f} \sum_{k=1}^{|y|} \log p_\theta(y^k \mid y^{<k}, x)$.

3 Method

3.1 A Unified Formulation of Loss Reweighting

In this section, we formulate loss reweighting into a unified framework, providing the basic optimization formulation of URGE, which is applicable across tasks and GU methods.

First, we introduce the notion of *forget units*. Given a reweighting granularity, we abstract heterogeneous forget targets as a set of units: $\mathcal{U}_f = \{u_i\}_{i=1}^M$. A forget unit corresponds to a training sample (image in image models, text sequence in LLMs) or a token in LLMs. Given any base forgetting loss $\ell_{\text{base}}(\theta; u_i)$ defined on a unit $u_i \in \mathcal{U}_f$, URGE assigns weight w_i for unit u_i and optimizes

$$\mathcal{L}_{\text{URGE}}(\theta) = \frac{1}{|\mathcal{U}_f|} \sum_{u_i \in \mathcal{U}_f} w_i \ell_{\text{base}}(\theta; u_i), \tag{2}$$

When the unit u_i is a sample (x_i, y_i), given the per-sample forgetting loss $\ell_{\text{forget}}(\theta; x_i, y_i)$, we have

$$\mathcal{L}_{\text{URGE}}^{(\text{sample})}(\theta) = \frac{1}{|\mathcal{D}_f|} \sum_{(x_i,y_i)\in\mathcal{D}_f} w_i \, \ell_{\text{forget}}(\theta; x_i, y_i), \tag{3}$$

where w_i is the weight for (x_i, y_i). For LLMs, the per-sample loss can be further reweighted at the token level, where each unit $u_{i,k}$ is the k-th target token y_i^k in the example (x_i, y_i). Taking GA as an example, the token-level reweighted loss is

$$\mathcal{L}_{\text{URGE}}^{(\text{token})}(\theta) = \frac{1}{Z} \sum_{(x_i,y_i)\in\mathcal{D}_f} \sum_{k=1}^{|y_i|} w_{i,k} \log p_\theta(y_i^k \mid y_i^{<k}, x_i), \tag{4}$$

where $w_{i,k}$ is the weight for $u_{i,k}$ and $Z = \sum_{(x_i,y_i)\in\mathcal{D}_f} |y_i|$ is the number of tokens.

Having formed a unified view of loss reweighting, the key question is how to compute weights. For each $u_i \in \mathcal{U}_f$,we decompose its weight into a static Conditional Representation Invariance (CRI) score $\text{CRI}(u_i)$, together with a dynamic Representation Similarity (RS) score $\text{RS}(u_i)$. $\text{CRI}(\cdot)$ focuses on "intrinsically critical units," up-weighting units strongly associated with the target knowledge; $\text{RS}(\cdot)$ focuses on "units that have not yet been fully forgotten," up-weighting units still strongly memorized by the model. Since $\text{CRI}(\cdot) \in [0, 1]$, we map it to a multiplicative factor bounded within $[\lambda_{\min}, \lambda_{\max}]$. With γ controlling the influence of RS, the weight of u_i at the tth unlearning step is

$$w_i^{(t)} = \left[\lambda_{\min} + \text{CRI}(u_i)(\lambda_{\max} - \lambda_{\min})\right] \cdot \left[\text{RS}^{(t)}(u_i)\right]^\gamma. \tag{5}$$

3.2 Conditional Representation Invariance (CRI)

For each unit $u_i \in \mathcal{U}_f$, $\text{CRI}(u_i)$ estimates the intrinsic degree to which u_i should be forgotten, reflecting how strongly it is associated with the target knowledge in θ_{ori}. $\text{CRI}(u_i)$ is static and can be computed offline before unlearning.

We first define the *condition* as the factor under which a unit's representation is formed, such as model parameters, inputs or contexts. We select the finest-grained condition C that is shared by all units under comparison while having the smallest sharing scope. We then apply a controlled perturbation $\mathcal{P}$ to the selected C, obtaining the perturbed condition $\widetilde{C} = \mathcal{P}(C)$. $\mathcal{P}$ can be random noise or semantic perturbation, depending on the scenario. Next, for each unit $u_i \in \mathcal{U}_f$, we compute its internal representations under the original and perturbed conditions using the original model θ_{ori}:

$$r_i^{(\ell)} = \mathcal{R}_{\theta_{\text{ori}}}^{(\ell)}(u_i; C), \qquad \widetilde{r}_i^{(\ell)} = \mathcal{R}_{\theta_{\text{ori}}}^{(\ell)}(u_i; \widetilde{C}), \tag{6}$$

where $\mathcal{R}_{\theta_{\text{ori}}}^{(\ell)}(\cdot)$ denotes the ℓth-layer representation extractor of the original model. Finally, we measure the similarity between the two representations and apply rank normalization within the comparison set $\mathcal{U}_f$ to obtain the CRI score, with function $\text{sim}(\cdot,\cdot)$ instantiated as cosine similarity:

$$\rho_i = \text{sim}\left(r_i^{(\ell)}, \widetilde{r}_i^{(\ell)}\right), \quad \text{CRI}(u_i) = \text{Norm}_{\mathcal{U}_f}(\rho_i). \tag{7}$$

A unit's representation contains target-related information and condition-specific cues from model state or input context. By perturbing this condition while keeping the unit and target unchanged, CRI measures the information that persists across conditions. A high CRI score thus indicates a stronger and more stable association with the target knowledge, making the unit more likely to carry semantically critical content to prioritize for forgetting.

Notably, CRI differs from conventional perturbation sensitivity: instead of perturbing the unit itself, it perturbs its external condition, thereby measuring the unit's dependence on external conditions rather than its intrinsic stability.

This design brings 3 advantages. (i) Representations are higher-dimensional than output distributions, offering information less redundant with the forgetting loss. (ii) The perturb-and-compare process brings an attributional view to better measure the intrinsic forgetting need. (iii) CRI is efficient and requires no retain data or external models.

3.2.1 Sample-level CRI.

When forget units are samples, we rank their CRI scores within the forget set $\mathcal{D}_f$, where the model parameters θ_{ori} serve as the finest-grained shared condition. Following our condition selection principle, we select the perturbation target $C = \theta_{\text{ori}}$. We instantiate the perturbation operator $\mathcal{P}$ as a controlled random Gaussian noise on θ_{ori}. The normalization is performed over the entire $\mathcal{D}_f$, and other operations follow the CRI definition.

3.2.2 TOKEN-LEVEL CRI.

For LLMs, we refine forget units to individual tokens and rank their CRI scores within each sample. Consider a question-answer sample (x_i, y_i), where x_i is the question prompt and $y_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,K_i})$ is the answer token sequence. These answer tokens share multiple forming conditions, including the global model parameters θ_{ori} and the question x_i. Compared with θ_{ori}, x_i is a more direct semantic condition that elicits and organizes the answer tokens, and it is shared only within this sample. Since tokens are scored and ranked within each sample, the finest-grained shared condition is sample-specific. Thus, for the i-th sample, we set the perturbation target $C_i = x_i$.

As the selected condition is a semantic input, we instantiate $\mathcal{P}$ as a semantic perturbation that replaces each sample-specific question x_i with a generic template $\tilde{x}_i$. For example, in TOFU benchmark, "What gender is author Basil Mahfouz Al-Kuwaiti?" is replaced with "Tell me about the author." Finally, the normalization is performed over each sample x_i, and other operations follow the CRI definition.

For text-only forget samples without an explicit question ($x_i = \varnothing$), we manually construct a sample-specific question x'_i as the original condition, $C'_i = x'_i$. All other steps remain unchanged: we use a generic question $\widetilde{x}'_i$ as the perturbed condition and compute the CRI score.

3.2.3 HIERARCHICAL TOKEN CRI.

For LLMs, a token's forgetting-priority depends on whether its corresponding sample is forgetting-critical within the forget set, and whether the token itself is forgetting-critical within the sample. Given a token $u_{i,t}$ from sample u_i, let $\mathrm{CRI}_{\text{sample}}(u_i)$ and $\mathrm{CRI}_{\text{token}}(u_{i,t})$ denote the sample- and token-level CRI scores, respectively. A basic approach is to treat all samples as equally critical, and directly use $\mathrm{CRI}_{\text{token}}(u_{i,t})$ to represent its forgetting priority. Beyond this, combining both levels leads to a hierarchical token CRI as $\mathrm{CRI}_{\text{hier}}(u_{i,t}) = \mathrm{CRI}_{\text{sample}}(u_i) \cdot \mathrm{CRI}_{\text{token}}(u_{i,t})$. Since the two scores are derived from conditions at different granularities, they provide distinct rather than redundant signals, enabling coarse-to-fine forgetting control.

3.3 REPRESENTATION SIMILARITY (RS)

For each unit $u_i \in \mathcal{U}_f$, RS score $\mathrm{RS}^{(t)}(u_i)$ estimates the extent to which u_i still needs to be forgotten at unlearning step t. Thus, it is dynamically updated online during unlearning.

Given a frozen original model θ_{ori} and the current t-step model $\theta^{(t)}$. For each u_i, we compute its ℓth-layer representations under the same original forming condition C_i:

$$r_{i,\text{ori}}^{(\ell)} = \mathcal{R}_{\theta_{\text{ori}}}^{(\ell)}(u_i; C_i), \qquad r_i^{(\ell,t)} = \mathcal{R}_{\theta^{(t)}}^{(\ell)}(u_i; C_i), \tag{8}$$

where $\mathcal{R}_\theta^{(\ell)}(\cdot)$ denotes the ℓth-layer representation extractor of model θ. We compute the representation similarity and map it to the RS score through a thresholded gate τ:

$$\mathrm{RS}^{(t)}(u_i) = \frac{\mathrm{ReLU}\left(\mathrm{sim}\left(r_i^{(\ell,t)}, r_{i,\text{ori}}^{(\ell)}\right) - \tau\right)}{1-\tau}. \tag{9}$$

A large $\mathrm{RS}^{(t)}(u_i)$ means that u_i's current representation remains close to its pre-unlearning counterpart, indicating that u_i's encoded memory remains under-forgotten; thus, we up-weight it to promote forgetting. Otherwise, u_i is down-weighted to avoid over-unlearning. Once the similarity falls below τ, $\mathrm{RS}^{(t)}(u_i)$ is directly truncated to 0, preventing further forgetting emphasis on this unit. This design adaptively adjusts the forgetting strength of each unit during unlearning and provides a unit-level early-stopping-like mechanism. The formulation applies at both sample and token levels.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Datasets and Models. For image classification, we consider both class-wise and random forgetting on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009), using AllCNN (Springenberg et al.,

Table 1: Class unlearning results on image classification. † indicates URGE-enhanced variants.

Model	ResNet-18, CIFAR-10				ViT, CIFAR-100			
Metric	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$
Original	$99.96_{\pm 0.00}$ (99.96)	$99.84_{\pm 0.00}$ (0.11)	$91.50_{\pm 0.00}$ (0.29)	33.45	$99.60_{\pm 0.00}$ (99.60)	$97.40_{\pm 0.00}$ (2.46)	$89.16_{\pm 0.00}$ (3.52)	35.19
Retrain	$0.00_{\pm 0.00}$ (0.00)	$99.95_{\pm 0.00}$ (0.00)	$91.79_{\pm 0.00}$ (0.00)	0.00	$0.00_{\pm 0.00}$ (0.00)	$99.86_{\pm 0.00}$ (0.00)	$92.68_{\pm 0.08}$ (0.00)	0.00
SatImp	$6.86_{\pm 1.20}$ (6.86)	$95.10_{\pm 0.87}$ (4.85)	$87.50_{\pm 1.03}$ (4.29)	5.33	$0.00_{\pm 0.00}$ (0.00)	$96.96_{\pm 0.72}$ (2.90)	$88.60_{\pm 0.90}$ (4.08)	2.33
WGA	$8.26_{\pm 2.23}$ (8.26)	$95.56_{\pm 0.72}$ (4.39)	$88.03_{\pm 0.99}$ (3.76)	5.47	$0.00_{\pm 0.00}$ (0.00)	$98.02_{\pm 0.50}$ (1.84)	$89.79_{\pm 0.57}$ (2.89)	1.58
LoReUn_s	$4.24_{\pm 0.22}$ (4.24)	$87.17_{\pm 2.43}$ (12.78)	$80.98_{\pm 3.03}$ (10.81)	9.28	$0.00_{\pm 0.00}$ (0.00)	$96.39_{\pm 1.04}$ (3.47)	$88.83_{\pm 1.29}$ (3.85)	2.44
LoReUn_d	$0.00_{\pm 0.00}$ (0.00)	$90.64_{\pm 1.92}$ (9.31)	$84.68_{\pm 2.17}$ (7.11)	5.47	$0.00_{\pm 0.00}$ (0.00)	$97.14_{\pm 0.98}$ (2.72)	$89.01_{\pm 1.11}$ (3.67)	2.13
GA	$4.86_{\pm 0.29}$ (4.86)	$85.82_{\pm 2.56}$ (14.13)	$79.77_{\pm 2.19}$ (12.02)	10.34	$0.00_{\pm 0.00}$ (0.00)	$95.62_{\pm 1.01}$ (4.24)	$87.66_{\pm 1.38}$ (5.02)	3.09
GA†	$0.91_{\pm 0.35}$ (0.91)	$92.40_{\pm 1.37}$ (7.55)	$84.43_{\pm 1.19}$ (7.36)	5.27	$0.00_{\pm 0.00}$ (0.00)	$98.33_{\pm 0.12}$ (1.53)	$90.04_{\pm 0.17}$ (2.64)	1.39
BS	$0.36_{\pm 0.09}$ (0.36)	$85.83_{\pm 1.81}$ (14.12)	$78.50_{\pm 2.00}$ (13.29)	9.26	$0.00_{\pm 0.00}$ (0.00)	$96.84_{\pm 0.21}$ (3.02)	$88.63_{\pm 0.23}$ (4.05)	2.36
BS†	$1.06_{\pm 0.44}$ (1.06)	$96.10_{\pm 0.85}$ (3.85)	$88.82_{\pm 0.96}$ (2.97)	2.63	$0.40_{\pm 0.00}$ (0.40)	$99.52_{\pm 0.04}$ (0.34)	$91.79_{\pm 0.12}$ (0.89)	0.54
SalUn	$0.32_{\pm 0.16}$ (0.32)	$90.14_{\pm 1.22}$ (9.81)	$84.22_{\pm 1.48}$ (7.57)	5.90	$0.00_{\pm 0.00}$ (0.00)	$98.60_{\pm 0.33}$ (1.26)	$90.92_{\pm 0.61}$ (1.76)	1.01
SalUn†	$0.28_{\pm 0.02}$ (0.28)	$95.02_{\pm 0.42}$ (4.93)	$87.19_{\pm 0.59}$ (4.60)	3.27	$0.00_{\pm 0.00}$ (0.00)	**99.70**$_{\pm 0.04}$ (0.16)	**92.16**$_{\pm 0.09}$ (0.52)	**0.23**
SCRUB	$0.00_{\pm 0.00}$ (0.00)	$99.06_{\pm 0.18}$ (0.89)	$89.39_{\pm 0.32}$ (2.40)	1.10	$0.00_{\pm 0.00}$ (0.00)	$99.22_{\pm 0.13}$ (0.64)	$91.87_{\pm 0.16}$ (0.81)	0.48
SCRUB†	$0.24_{\pm 0.00}$ (0.24)	**99.86**$_{\pm 0.02}$ (0.09)	**91.53**$_{\pm 0.05}$ (0.26)	**0.20**	$0.00_{\pm 0.00}$ (0.00)	$99.64_{\pm 0.05}$ (0.22)	$92.09_{\pm 0.18}$ (0.59)	0.27

Table 2: Class unlearning results on Diffusion Models. † indicates URGE-enhanced variants.

Model	DDPM, CIFAR-10										SD, Imagenette									
Forget Class	airplane		automobile		cat		dog		ship		trash truck		tench		tape player		chain saw		parachute	
	UA ↓	FID ↓	UA ↓	FID ↓	UA ↓	FID ↓	UA ↓	FID ↓	UA ↓	FID ↓	UA ↓	FID ↓	UA ↓	FID ↓	UA ↓	FID ↓	UA ↓	FID ↓	UA ↓	FID ↓
original	97.8	20.96	100.0	22.41	95.0	22.40	98.4	21.59	100.0	21.94	84.0	9.16	90.0	11.39	87.0	9.90	100.0	8.12	100.0	9.74
ESD	0.40	34.94	0.60	19.58	0.60	23.38	0.60	22.68	0.00	24.90	0.00	10.09	0.00	13.18	0.00	08.42	0.00	09.89	0.00	06.58
FMN	0.60	23.16	0.00	22.75	0.40	20.01	0.00	21.73	0.60	20.12	0.02	10.82	0.00	07.23	0.00	07.30	0.00	07.34	0.00	08.86
LoReUn_s	0.60	23.20	0.20	20.25	0.00	21.68	0.00	20.10	0.00	21.07	0.00	11.92	0.00	09.12	0.00	08.45	0.02	08.45	0.00	08.30
LoReUn_d	0.60	23.60	0.00	20.63	0.00	21.03	0.60	19.68	0.20	19.94	0.00	12.33	0.00	08.41	0.00	07.98	0.02	07.40	0.00	08.45
RL	0.00	22.82	0.00	21.41	0.20	21.64	0.00	21.28	0.00	21.16	0.00	12.10	0.00	07.89	0.00	07.74	0.18	07.97	0.00	08.62
RL†	0.00	**21.30**	0.00	20.23	0.00	20.84	0.00	19.99	0.00	18.54	0.00	**10.05**	0.00	07.24	0.00	**07.09**	0.00	07.17	0.00	07.96
SalUn	0.00	24.62	0.20	19.98	0.40	20.49	0.00	21.35	0.80	18.98	0.00	11.73	0.00	07.26	0.00	08.40	0.00	07.79	0.00	08.06
SalUn†	0.00	24.17	0.00	**19.70**	0.20	**18.82**	0.00	**16.30**	0.00	**17.98**	0.00	10.64	0.00	**07.01**	0.00	07.50	0.00	**07.14**	0.00	**06.97**

2014), ResNet-18 (He et al., 2016), and ViT (Dosovitskiy, 2020). For image generation, we evaluate class-wise forgetting on CIFAR-10 with DDPM (Ho et al., 2020) and on ImageNette (Howard & Gugger, 2020) with Stable Diffusion (SD) (Rombach et al., 2022), where the forget target is specified by a class label in DDPM and by a class-name text prompt in SD. For LLMs, we evaluate URGE on TOFU (Maini et al., 2024) with Llama-3.2-1B (Grattafiori et al., 2024) and Llama-2-7B (Touvron et al., 2023), on WMDP (Li et al., 2024) with Zephyr-7B-BETA (Tunstall et al., 2023), and on MUSE (Shi et al., 2025) with Llama-2-7B. See Appendix D for extended experiments.

Baselines. For image classification, we consider four GU baselines: gradient ascent (GA), boundary shrink (BS) (Chen et al., 2023), Saliency Unlearning (SalUn), and SCRUB (Kurmanji et al., 2023); and plug URGE into each of them. We also compare with four GA-based reweighting baselines: SatImp, Weighted GA (WGA), LoReUn-static and LoReUn-dynamic (Li et al., 2025a). For diffusion models, we consider four GU baselines: Random Labeling (RL), SalUn, Erased Stable Diffusion (ESD) (Gandikota et al., 2023), and Forget-Me-Not (FMN) (Zhang et al., 2024a); and plug URGE into RL and SalUn. We further include two RL-based reweighting baselines: LoReUn-static and LoReUn-dynamic. For LLMs, we compare against three GU baselines: GA, Negative Preference Optimization (NPO) (Zhang et al., 2024b), Representation Misdirection for Unlearning (RMU) (Li et al., 2024); and plug URGE into them. We also include three GA-based reweighting baselines: SatImp, WGA, and Weighted Token-wise NPO (WTNPO) (Wang et al., 2025c)

Evaluation Metrics. For image classification, we report unlearning accuracy (UA), retain accuracy (RA), and test accuracy (TA) on $\mathcal{D}_f$, $\mathcal{D}_r$, and the test set $\mathcal{D}_t$, respectively. We report the mean and standard deviation averaged over 5 runs. We calculate the gap of each metric relative to the Retrain oracle, and report $\overline{\Delta}$ as the mean gap across all three metrics. For DMs, we use an external classifier (ResNet-34 for DDPM and ResNet-50 for SD) to measure UA on the forget target, and use FID to evaluate the generation quality on the retain set. For LLMs, we adopt benchmark-specific evaluation. On TOFU, we report Extraction Strength (ES) (Wang et al., 2025b) on both the forget and retain sets, together with Forget Quality (FQ) and Model Utility (MU).

Table 3: LLM unlearning results on TOFU. † indicates URGE-enhanced variants.

Model	Llama3.2-1B								Llama2-7B							
Setup	Forget-1%				Forget-5%				Forget-1%				Forget-5%			
Metric	ES-u ↓	ES-r ↑	MU ↑	FQ ↑	ES-u ↓	ES-r ↑	MU ↑	FQ ↑	ES-u ↓	ES-r ↑	MU ↑	FQ ↑	ES-u ↓	ES-r ↑	MU ↑	FQ ↑
original	0.7521	0.7347	0.6033	0.0100	0.7265	0.7347	0.5989	1.3e-13	1.0000	0.9882	0.6276	1.3e-3	0.9808	0.9882	0.6341	5.9e-14
retrain	0.0693	0.7511	0.6054	1.0000	0.0629	0.7452	0.6008	1.0000	0.0680	0.9849	0.6282	1.0000	0.0671	0.9861	0.6334	1.0000
SatImp	0.0875	0.2551	0.5681	0.0143	0.1133	0.2261	0.5238	2.6e-5	0.1121	0.7027	0.5783	0.2149	0.3064	0.6720	0.5821	9.6e-10
WGA	0.0779	0.2626	0.5669	0.0286	0.1098	0.2239	0.5216	6.6e-5	0.1033	0.6814	0.5766	0.2906	0.1531	0.4801	0.5863	4.6e-7
WTNPO	0.0522	0.3683	0.5637	0.1650	0.0913	0.2085	0.5409	9.8e-6	0.1297	0.7485	0.5843	0.1009	0.1241	0.4091	0.5537	1.1e-7
GA	0.0738	0.2484	0.5352	0.4046	0.2763	0.4462	0.5656	6.5e-13	0.1846	0.9173	0.6002	0.0143	0.0320	0.2663	0.5011	7.2e-10
GA†	0.0537	0.2359	0.5410	0.9900	0.1025	0.2302	0.5477	1.8e-4	0.1742	0.9381	0.6207	0.0486	0.2141	0.6496	0.5961	1.3e-8
NPO	0.0548	0.5323	0.5849	0.9188	0.1032	0.2104	0.5255	0.1123	0.0598	0.4862	0.5872	0.7659	0.1052	0.4829	0.5729	0.1779
NPO†	0.0711	0.5099	0.5844	**0.9900**	0.0979	0.2679	0.5330	**0.1521**	0.0555	0.5417	0.5759	**0.9900**	0.1129	0.5515	0.5701	**0.2435**
RMU	0.0345	0.4903	0.5723	0.9188	0.0361	0.5022	0.5622	6.7e-7	0.1065	0.6355	0.6020	0.2657	0.0286	0.3382	0.5949	0.0006
RMU†	0.0361	0.4968	0.5779	0.9900	0.0327	0.5337	0.5774	4.6e-6	0.0987	0.6771	0.6205	0.4046	0.0819	0.4319	0.6092	0.0021

4.2 MU IN DIFFERENT TASKS

We evaluate URGE across 3 representative tasks: image classification, image generation, and text generation. The configurations are as follows: (i) For image classification, we randomly select the target class with index 1 ('Automobile' in CIFAR-10 and 'Aquarium Fish' in CIFAR-100). Reweighting is performed at the sample level using representations before the fully-connected classifier. (ii) For image generation, we evaluate five target classes across two dataset-model settings. Sample-level reweighting is applied using representations extracted from the pre-output layer of the diffusion U-Net before the final convolution. (iii) For text generation, all reweighting is token-level, and URGE uses its basic rather than hierarchical variant. We use the outputs of the final Transformer layer as representations, which serve as high-level semantic features before the LM head.

Tables 1, 2, and 3 report the results across tasks. URGE generally improves the forgetting-retention trade-off of its corresponding GU methods across datasets, architectures, and forgetting objectives, showing benefits beyond the original optimization objective. This improvement is reflected in task-specific metrics, primarily through better retain utility (RA, FID, ES-r, and MU) under comparable forgetting strength (UA and ES-u). FQ and $\overline{\Delta}$ provide unified measures that directly demonstrate such improvement. Besides, existing reweighting baselines generally outperform vanilla GU methods, but remain inferior to URGE in most settings, showing the advantage of URGE.

4.3 ABLATION

We conduct ablation studies on each component in URGE. Table 4a reports the results on image classification and generation task, showing that each component contributes to the final performance, and their combination yields optimal complementary effects (e.g. $1.96\times$ reduction in $\overline{\Delta}$). For LLMs, we evaluate sample-, token-level and also hierarchical token-level CRI/RS, using each sample's last token representation as the sample representation. As shown in Table 4b, The effects of CRI and RS vary with the reweighting granularity and favor different sides of the forgetting–retention trade-off; When used individually, RS generally yields higher FQ, whereas CRI tends to improve MU. Their combination, however, consistently achieves the best FQ of 0.99 across granularities. Overall, using the same component, finer-grained reweighting often yields better performance.

4.4 HYPERPARAMETER ANALYSIS

We conduct a hyperparameter analysis by varying 3 factors: the RS threshold τ, the RS exponent γ and the representation layer choice. Figure 2 shows that moderately increasing τ improves URGE, but setting it too large degrades the results, with $\tau = 0.88$ being the most suitable and achieving $FQ = 0.99$. URGE's performance remains relatively stable to γ, increasing γ generally strengthens forgetting-utility trade-off, and values around 3.0–3.5 provide the best overall performance. Despite these variations, URGE remains robust without dedicated tuning. Besides, using representations from deeper hidden layers yields better performance, whereas using the final logits from the output projection layer leads to a substantial drop.

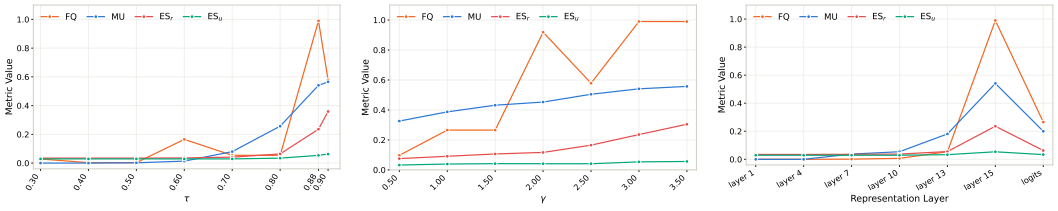
Table 4: Ablation study of CRI and RS on three tasks.

(a) Ablation on image classification and generation.

CRI	RS	Image Classification				Image Generation	
		UA ↓	RA ↑	TA ↑	$\overline{\Delta}$ ↓	UA ↓	FID ↓
		GA, RN18, CIFAR10				*RL, DDPM, CIFAR10*	
-	-	4.86	85.82	79.77	10.34	0.00	22.82
✓	-	6.86	91.22	84.19	7.73	0.00	21.82
-	✓	1.12	91.68	84.21	5.66	0.00	21.49
✓	✓	0.91	92.40	84.43	**5.27**	0.00	**21.30**
		GA, VIT, CIFAR10				*RL, SD, Imagenette*	
-	-	0.32	96.00	94.58	2.83	0.00	12.10
✓	-	0.30	98.00	96.58	1.49	0.00	11.56
-	✓	0.00	98.85	97.46	0.81	0.00	10.80
✓	✓	0.00	99.87	98.49	**0.13**	0.00	**10.05**

(b) Ablation study for text generation.

GA, Llama3.2-1B, Forget-1%						
	CRI	RS	ES-u ↓	ES-r ↑	MU ↑	FQ ↑
baseline (GA)	-	-	0.0738	0.2484	0.5352	0.4046
sample-level	✓	-	0.1005	0.3360	0.5569	0.4046
	-	✓	0.1085	0.2540	0.5448	0.2657
	✓	✓	0.0677	0.1613	0.4800	**0.9900**
token-level	✓	-	0.0922	0.3411	0.5570	0.4046
	-	✓	0.0754	0.3014	0.5446	0.7659
	✓	✓	0.0537	0.2359	0.5410	**0.9900**
hierarchical token	✓	-	0.0402	0.1245	0.4371	0.7659
	✓	✓	0.0690	0.2327	0.5440	**0.9900**



(a) RS threshold τ. (b) RS exponent γ. (c) Representation layer choice.

Figure 2: Hyperparameter Analysis on Llama3.2-1B, TOFU-Forget 1%, GA†.

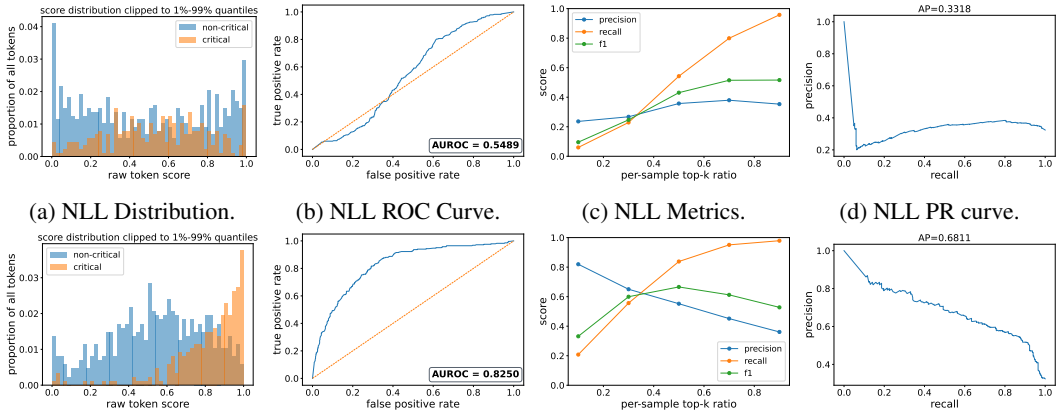
4.5 Mechanism Analysis

CRI Scores vs. LLM-as-Judge Annotations. To evaluate whether CRI captures tokens' forgetting criticality, we use GPT-5.5-Pro as an external LLM-as-Judge to annotate answer tokens in the forget set as critical or non-critical based on their semantic relevance to the target knowledge. We compare CRI with NLL, a representative output-distribution-based signal used by baselines such as SatImp and WGA, where lower NLLs indicate higher criticality. Figure 3 shows that CRI produces more distinguishable score distributions for critical and non-critical tokens. The ROC curves confirm the stronger ranking capability of CRI, improving AUROC from 0.5489 to 0.8250. Figures 3c, 3g show the trends of critical token identification performance (Precision, Recall, and F1) when selecting different top-K ratios of tokens. CRI generally surpasses NLL across selection ratios.

Representation Similarity Dynamics. We compare GA's RS trajectories with and without RS reweighting, by computing the similarity between a unit's original and current representation at each unlearning step. At the token level, we trace all tokens' RS within one random sample; at the sample level, we randomly trace 20 samples. Figures 4 show that: (i) RS up-weights slowly forgotten units, accelerating the forgetting of resistant units and the overall process. (ii) RS down-weights rapidly forgotten units, and further enables unit-level early stopping once their representations have sufficiently shifted, mitigating over-unlearning. Such dynamical adjustment mitigates GA's uneven forgetting progress, leading to a more balanced unlearning process.

4.6 Complexity Analysis

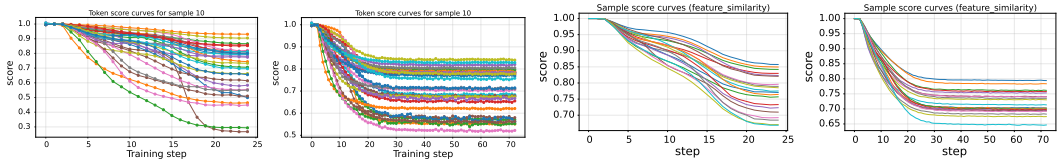
Let $N = |\mathcal{D}_f|$, Z_i be the number of answer tokens in sample i, $Z = \sum_{i=1}^{N} Z_i$, h be the representation dimension, and $\mathcal{F}_{\mathcal{D}_f}^{(l)}$ be the cost of one forward pass over $\mathcal{D}_f$ up to layer l. Ignoring perturbation construction, sample-level CRI costs $\mathcal{O}(2\mathcal{F}_{\mathcal{D}_f}^{(l)} + Nh + N \log N)$, where $2\mathcal{F}_{\mathcal{D}_f}^{(l)}$ accounts for representation extraction under the original and perturbed conditions, Nh for cosine similarity between N pairs of representations, and $N \log N$ for rank normalization across $\mathcal{D}_f$ samples. Token-level CRI similarly costs $\mathcal{O}(2\mathcal{F}_{\mathcal{D}_f}^{(l)} + Zh + \sum_{i=1}^{N} Z_i \log Z_i)$, with similarity computed over all Z tokens and normalization performed within each sample. Hierarchical CRI shares the original-condition for-



(a) NLL Distribution. (b) NLL ROC Curve. (c) NLL Metrics. (d) NLL PR curve.

(e) CRI Score Distribution. (f) CRI ROC Curve. (g) CRI Metrics. (h) CRI PR curve.

Figure 3: NLL- and CRI-based scores against token-level LLM-as-judge annotations, using Llama3.2-1B, TOFU-1%.



(a) Token-level, GA. (b) Token-level, GA+RS. (c) Sample-level, GA. (d) Sample-level, GA+RS.

Figure 4: RS curves on Llama3.2-1B, TOFU-1%.

Table 5: Runtime for the baseline (T_{base}), offline CRI (T_{CRI}), and baseline with online RS ($T_{\text{base+RS}}$).

Setting	T_{base}	T_{CRI}	$T_{\text{base+RS}}$	T_{total}
GA, ResNet-18, CIFAR-10	15.23	10.84	15.47	26.31
RL, DDPM, CIFAR-10	451.40	9.29	569.80	579.09
GA, TOFU-1%, Llama3.2-1B	218.34	5.40	298.55	303.95

ward pass between the two levels, resulting in $\mathcal{O}(3\mathcal{F}^{(l)}_{\mathcal{D}_f} + (N+Z)h + N\log N + \sum_{i=1}^{N} Z_i \log Z_i)$. During unlearning, RS reuses representations from the standard forward pass and compares them with cached original references, adding only $\mathcal{O}(\sum_{e=1}^{E} U_e h)$ over E unlearning steps.

We compare the runtime of URGE's components with their baselines. As shown in Table 5, offline CRI is lightweight (5-11s), and online RS remains efficient despite scaling with epochs.

5 Conclusion

This work proposed URGE, a unified representation-guided reweighting framework for GU. Requiring no retain data or external models, URGE is applicable across diverse tasks and GU methods. URGE highlights the potential of internal representations as practical signals for adaptive and fine-grained forgetting control. Using Conditional Representation Invariance and Representation Similarity, URGE estimates the intrinsic and residual forgetting needs of samples/tokens. Extensive experiments validate its effectiveness and broad applicability. We expect this work to motivate future research on more semantic and fine-grained unlearning control.

6 AI USE STATEMENT

In this work, we used generative AI tools for providing feedback on research methodology or experiments, implementing methods, assisting with translation. We have not used generative AI tools for designing research methodology or experiments, cleaning and reformatting dataset, support qualitative and thematic data analysis. Generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses are not applicable to this work. Additionally, we used generative AI tools for creating or editing software code, sourcing/searching for information, editing a research paper to improve readability, identifying relevant literature, suggesting a structure for a research paper, and interpreting results. We have reviewed all AI-assisted work. All AI-generated suggestions on research methodology, experiments, paper structure, and readability were manually examined by the authors and used only as auxiliary feedback rather than as final decisions. AI-assisted code was reviewed, debugged, and tested by the authors to ensure correctness and consistency with our intended implementation. For literature identification and information searching, we verified the relevance and accuracy of the suggested references by checking the original papers and sources. All final experimental designs, result interpretations, and conclusions were independently determined and validated by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

7 REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our results. The problem formulation and background are described in Section 2, and all methods details are described in Section 3, while the key algorithmic details are presented in Appendix A. The experimental setup, including datasets, model architectures, unlearning methods, evaluation metrics, and implementation settings, is reported in Section 4.1. Additional details are provided in Appendix C. We will also include anonymized source code, configuration files, and scripts in the supplementary materials to facilitate independent reproduction of the main experimental results.

REFERENCES

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, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7766–7775, 2023.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7210–7217, 2023.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. *Advances in Neural Information Processing Systems*, 38:1540–1567, 2026.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2426–2436, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jeremy Howard and Sylvain Gugger. Fastai: a layered api for deep learning. *Information*, 11(2): 108, 2020.

Seunghee Koh, Sunghyun Baek, Youngdong Kim, and Junmo Kim. Forget what matters, keep the rest: Selective unlearning of informative tokens. *arXiv preprint arXiv:2604.17785*, 2026.

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

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36:1957–1987, 2023.

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.

Xiang Li, Qianli Shen, Haonan Wang, and Kenji Kawaguchi. Loreun: Data itself implicitly provides cues to improve machine unlearning. *arXiv preprint arXiv:2507.22499*, 2025a.

Zitong Li, Qingqing Ye, and Haibo Hu. Funu: Boosting machine unlearning efficiency by filtering unnecessary unlearning. In *Proceedings of the ACM on Web Conference 2025*, pp. 3366–3376, 2025b.

Ruixuan Liu, Li Xiong, et al. Direct token optimization: A self-contained approach to large language model unlearning. *arXiv preprint arXiv:2510.00125*, 2025.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

Peizhi Niu, Evelyn Ma, Huiting Zhou, Duo Zhou, Huan Zhang, S Rasoul Etesami, and Olgica Milenkovic. Guard: Guided unlearning and retention via data attribution for large language models. *arXiv preprint arXiv:2506.10946*, 2025.

Soumyadeep Pal, Changsheng Wang, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. Llm unlearning reveals a stronger-than-expected coreset effect in current benchmarks. *arXiv preprint arXiv:2504.10185*, 2025.

Vaidehi Patil, Elias Stengel-Eskin, and Mohit Bansal. Upcore: Utility-preserving coreset selection for balanced unlearning. *arXiv preprint arXiv:2502.15082*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. In *International Conference on Learning Representations*, volume 2025, pp. 27797–27818, 2025.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Toan Tran, Ruixuan Liu, and Li Xiong. Tokens for learning, tokens for unlearning: Mitigating membership inference attacks in large language models via dual-purpose training. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22872–22888, 2025.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

Yixin Wan, Anil Ramakrishna, Kai-Wei Chang, Volkan Cevher, and Rahul Gupta. Not every token needs forgetting: Selective unlearning to limit change in utility in large language model unlearning. *arXiv preprint arXiv:2506.00876*, pp. 622–632, 2025.

Lingzhi Wang, Xingshan Zeng, Jinsong Guo, Kam-Fai Wong, and Georg Gottlob. Selective forgetting: Advancing machine unlearning techniques and evaluation in language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 843–851, 2025a.

Qizhou Wang, Bo Han, Puning Yang, Jianing Zhu, Tongliang Liu, and Masashi Sugiyama. Towards effective evaluations and comparisons for llm unlearning methods. In *International Conference on Learning Representations*, volume 2025, pp. 23327–23355, 2025b.

Qizhou Wang, Jin Zhou, Saebyeol Shin, Bo Han, Kilian Weinberger, et al. Rethinking llm unlearning objectives: A gradient perspective and go beyond. In *International Conference on Learning Representations*, volume 2025, pp. 61897–61931, 2025c.

Yaxuan Wang, Jiaheng Wei, Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Parag Shah, Yujia Bao, Yang Liu, and Wei Wei. Llm unlearning via loss adjustment with only forget data. In *International Conference on Learning Representations*, volume 2025, pp. 43076–43104, 2025d.

Jiawei Wu and Doudou Zhou. Unlearning what matters: Token-level attribution for precise language model unlearning. *arXiv preprint arXiv:2605.00364*, 2026.

Jing Wu, Trung Le, Munawar Hayat, and Mehrtash Harandi. Erasing undesirable influence in diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28263–28273, 2025.

Puning Yang, Qizhou Wang, Zhuo Huang, Tongliang Liu, Chengqi Zhang, and Bo Han. Exploring criteria of loss reweighting to enhance llm unlearning. *arXiv preprint arXiv:2505.11953*, 2025.

Zhengbang Yang, Yisheng Zhong, Junyuan Hong, and Zhuangdi Zhu. Catnip: Llm unlearning via calibrated and tokenized negative preference alignment. *arXiv preprint arXiv:2602.02824*, 2026.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 8403–8419, 2024.

Liheng Yu, Zhe Zhao, Yuxuan Wang, Pengkun Wang, Xiaofeng Cao, Binwu Wang, and Yang Wang. Falw: A forgetting-aware loss reweighting for long-tailed unlearning. *arXiv preprint arXiv:2601.18650*, 2026.

Gizem Yüce, Giorgos Nikolaou, and Nicolas Flammarion. Learning what to forget: Improving llm unlearning via learned token-level importance. *arXiv preprint arXiv:2606.06320*, 2026.

Naixin Zhai, Pengyang Shao, Binbin Zheng, Yonghui Yang, Fei Shen, Long Bai, and Xun Yang. Maximizing local entropy where it matters: Prefix-aware localized llm unlearning. *arXiv preprint arXiv:2601.03190*, 2026.

Gong Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not: Learning to forget in text-to-image diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1755–1764, 2024a.

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

Guoshenghui Zhao, Huawei Lin, and Weijie Zhao. Rapidun: Influence-driven parameter reweighting for efficient large language model unlearning. *arXiv preprint arXiv:2512.04457*, 2025.

Xiangyu Zhou, Yao Qiang, Saleh Zare Zade, Douglas Zytko, Prashant Khanduri, and Dongxiao Zhu. Not all tokens are meant to be forgotten. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 38173–38182, 2026.

Appendix

A Pseudo Codes

The overall procedure of URGE is summarized in Algorithm 1. URGE first estimates the intrinsic forgetting need of each forget unit offline using Conditional Representation Invariance (CRI). During unlearning, it dynamically measures the residual forgetting need via Representation Similarity (RS) and combines the two signals to adaptively reweight the base forgetting loss. The retain set $\mathcal{D}_r$ is optional and is used only when required by the underlying GU method, rather than for weight construction.

Algorithm 1 URGE: Unified Representation-Guided Reweighting Framework for Gradient-based Unlearning

Input: Original model θ_o, forget units $\mathcal{U}_f = \{u_i\}_{i=1}^{M}$, base forgetting loss ℓ_{base}, retain set $\mathcal{D}_r$ (optional), forming conditions C_i, layer index l, epoch number E, learning rate η, $\lambda_{\min}, \lambda_{\max}, \gamma, \tau$, λ (retain weight, used only if use_retain), perturbation operator $\mathcal{P}$, $use_retain \in \{\text{True}, \text{False}\}$
Output: Unlearned model θ_u
– Phase 1: Offline Computation: Conditional Representation Invariance (CRI) –
for $u_i \in \mathcal{U}_f$ **do**
    Extract original representation $r_i \leftarrow \mathcal{R}^{(l)}_{\theta_o}(u_i; C_i)$
    Apply perturbation to the condition $\tilde{C}_i \leftarrow \mathcal{P}(C_i)$
    Extract representation under perturbed condition $\tilde{r}_i \leftarrow \mathcal{R}^{(l)}_{\theta_o}(u_i; \tilde{C}_i)$
    Compute similarity $\rho_i \leftarrow \text{cosine_sim}(r_i, \tilde{r}_i)$
end for
Rank normalize across comparison set: $I_i \leftarrow \text{rank_normalize}(\rho_i, \mathcal{U}_f)$
Cache original representations for online phase: $r_i^{\text{ori}} \leftarrow \mathcal{R}^{(l)}_{\theta_o}(u_i; C_i)$
– Phase 2: Online Computation via Representation Similarity (RS), and Reweighted Unlearning –
Initialize unlearned model $\theta^0 \leftarrow \theta_o$
for $e = 0$ to $E - 1$ **do**
    Initialize forgetting loss $\mathcal{L}_{\text{URGE}} \leftarrow 0$
    for $u_i \in \mathcal{U}_f$ **do**
        Extract current representation $r_i^{\text{cur}} \leftarrow \mathcal{R}^{(l)}_{\theta^e}(u_i; C_i)$
        Compute similarity to original: $s \leftarrow \text{cosine_sim}(r_i^{\text{cur}}, r_i^{\text{ori}})$
        Compute RS score: $S_i \leftarrow \max(0, s - \tau)/(1 - \tau)$
        Compute adaptive weight: $w_i \leftarrow (\lambda_{\min} + I_i(\lambda_{\max} - \lambda_{\min})) \cdot S_i^{\gamma}$
        Accumulate weighted forgetting loss: $\mathcal{L}_{\text{URGE}} \leftarrow \mathcal{L}_{\text{URGE}} + \frac{1}{|\mathcal{U}_f|} w_i \ell_{\text{base}}(\theta^e; u_i)$
    end for
    if use_retain **then**
        Compute retain loss: $\mathcal{L}_{\text{Retain}} \leftarrow \mathcal{L}_{\text{task}}(\theta^e; \mathcal{D}_r)$
        Merge losses: $\mathcal{L}_{\text{Total}} \leftarrow \mathcal{L}_{\text{URGE}} + \lambda \mathcal{L}_{\text{Retain}}$
    else
        Set total loss: $\mathcal{L}_{\text{Total}} \leftarrow \mathcal{L}_{\text{URGE}}$
    end if
    Compute gradients $\nabla_{\theta^e} \mathcal{L}_{\text{Total}}$
    Update model parameters: $\theta^{e+1} \leftarrow \theta^e - \eta \cdot \nabla_{\theta^e} \mathcal{L}_{\text{Total}}$
end for
Obtain final model $\theta_u = \theta^E$
return θ_u

B Analysis: CRI v.s. Output-Distribution-Based Scores

Existing token-level reweighting methods derive weights from prediction-related statistics, including token likelihood, relative likelihood change, and predictive entropy. However, these signals are not independent from the original forgetting objective, as their weighting functions can be reconstructed from the token-level loss or its gradient.

For a target token y_t, we denote its predicted probability as $q_t = p_\theta(y_t|c_t)$ and the token-level loss as $\ell_t = -\log q_t$. Existing methods construct weights from different prediction statistics:

$$\begin{aligned} w_t^{\mathrm{WGA}} &= q_t^{\alpha}, \\ w_t^{\mathrm{SatImp}} &= q_t^{\beta_1}(1-q_t)^{\beta_2}, \\ w_t^{\mathrm{WTNPO}} &= \frac{2q_t^{\beta}}{q_t^{\beta} + (q_t^0)^{\beta}}, \\ w_t^{\mathrm{Entropy}} &= H(p_t) = -\sum_v p_t(v)\log p_t(v), \end{aligned} \tag{10}$$

where $q_t^0 = p_{\theta_0}(y_t|c_t)$ denotes the probability from the reference model in WTNPO, and p_t denotes the full predictive distribution.

For likelihood-based methods, the weighting signals can be directly rewritten using the original token loss. Since

$$q_t = e^{-\ell_t}, \tag{11}$$

the weights of WGA and SatImp become

$$\begin{aligned} w_t^{\mathrm{WGA}} &= e^{-\alpha\ell_t}, \\ w_t^{\mathrm{SatImp}} &= e^{-\beta_1\ell_t}(1-e^{-\ell_t})^{\beta_2}. \end{aligned} \tag{12}$$

Similarly, let $\ell_t^0 = -\log q_t^0$ denote the reference-model loss. Then WTNPO can be rewritten as

$$w_t^{\mathrm{WTNPO}} = \frac{2e^{-\beta\ell_t}}{e^{-\beta\ell_t} + e^{-\beta\ell_t^0}}. \tag{13}$$

Therefore, WGA, SatImp, and WTNPO construct their weights directly from token-level loss values (or relative loss changes), rather than introducing an independent information source.

Entropy-based weighting considers the full predictive distribution instead of only the target-token probability. However, the predictive distribution is already encoded in the original CE gradient:

$$\nabla_{z_t}\ell_t = p_t - e_{y_t}, \tag{14}$$

where z_t denotes the output logits. Therefore,

$$p_t = \nabla_{z_t}\ell_t + e_{y_t}, \tag{15}$$

and the entropy can also be reconstructed from the original logit gradient:

$$H(p_t) = -\sum_v (\nabla_{z_t}\ell_t + e_{y_t})_v \log(\nabla_{z_t}\ell_t + e_{y_t})_v. \tag{16}$$

Thus, entropy-based weighting also relies on information already contained in the original optimization signal.

Overall, although existing methods exploit different prediction statistics, including likelihood, relative likelihood change, and entropy, their weighting signals can all be derived from the original loss or gradient. Therefore, these methods mainly transform existing optimization signals rather than introducing independent token-level information.

C Experiment Settings

The experiments are implemented in Python 3.9 with eight NVIDIA GeForce RTX 4090. For image classification task, we validate URGE and baselines on two datasets using three networks. The training and retraining processes share the same hyperparameters with respect to the same dataset and model. For easy reproduction, Table C1 lists the details. '$-$' indicates that the corresponding hyperparameter is not used. Note that all ViT configurations use a timm ViT-B/16 pretrained on ImageNet-21K and fine-tuned on ImageNet-1K before CIFAR-10/100 fine-tuning, with CIFAR images bilinearly upsampled from 32×32 to 224×224 and normalized to match the pretrained model's input scale.

Table C1: Training hyperparameters for original and retrain model in image classification tasks.

Dataset	Model	Batch size	Optimizer	lr	momentum	lr decay step	gamma	wd	epoch
CIFAR-10	AllCNN	64	SGD	0.1	0.9	-	-	-	30
	ResNet18	128	SGD	0.01	0.9	-	-	0.0005	200
	ViT	64	SGD	0.01	0.9	-	-	-	10
CIFAR-100	AllCNN	128	Adam	0.001	-	10	0.0001	-	300
	ResNet18	64	SGD	0.01	0.9	-	-	0.0005	200
	ViT	64	SGD	0.01	0.9	-	-	-	10

We evaluate class-wise unlearning on CIFAR-10 DDPMs and Stable Diffusion v1.4 on Imagenette. For DDPM, we use a 1,000-step conditional diffusion model and fine-tune the pretrained model for 1,000 iterations with Adam (10^{-4}); evaluation generates 500 images per class using 1,000 denoising steps and guidance scale 2.0. For Stable Diffusion, all models are trained for five epochs at 512×512 with learning rate 10^{-5}. For evaluation, we use LMS sampling for 100 steps and guidance scale 7.5.

For LLM unlearning experiments on TOFU, we first fine-tune Llama-3.2-1B-Instruct and Llama-2-7B-chat models on the full TOFU dataset to obtain the corresponding full models. Both models are trained using supervised fine-tuning with the same optimization configuration. Specifically, we use the paged AdamW 32-bit optimizer with a learning rate of 1×10^{-5}, a weight decay of 0.01, and a linear warmup and decay learning rate schedule. The models are trained for 5 epochs with 1 warmup epoch. Training is conducted on 2 GPUs with a per-device batch size of 4 and gradient accumulation steps of 4, resulting in an effective batch size of 32. We use BF16 precision, FlashAttention-2, and gradient checkpointing to improve training efficiency. All experiments are performed with DeepSpeed ZeRO Stage-3 distributed training and a maximum sequence length of 512 tokens. The random seed is set to 0 for reproducibility.

For LLM unlearning experiments on MUSE, we directly use the officially released target models from the MUSE benchmark. Specifically, we adopt the Llama-2-7B-based target checkpoints provided by MUSE. These models are used as the starting checkpoints for unlearning evaluation following the official MUSE protocol. The target models are based on BF16 precision and a maximum context length of 2048 tokens. On MUSE, we report Verbatim Memorization (VerbMem), Knowledge Memorization (KnowMem), Privacy Leakage (PrivLeak), and Utility Preservation (UtilPres).

For LLM unlearning experiments on WMDP, we use the Zephyr-7B-BETA model as the target model, following the standard evaluation protocol of the WMDP benchmark using accuracy on multiple-choice question answering.

D Extended Experiment Results

D.1 Extended Results for Image Classification Unlearning

We provide additional image classification results to evaluate URGE under a broader range of architectures, datasets, forgetting targets, and unlearning protocols. Specifically, we extend the class-unlearning evaluation to AllCNN, ResNet-18, and ViT on CIFAR-10 and CIFAR-100, examine random sample unlearning with different forget ratios, and further consider the more challenging sequential unlearning setting in which multiple classes are removed successively. We also report membership inference attack results to assess whether the observed forgetting translates into improved privacy protection.

Extended Class Unlearning Results

For class unlearning, Tables D2 and D3 show that incorporating URGE generally moves the base GU methods closer to the retraining oracle. The improvement is particularly pronounced for GA, SalUn, and SCRUB: URGE substantially preserves RA and TA while maintaining a low UA, indicating that its gains do not arise merely from applying stronger and potentially destructive forgetting updates. For example, on AllCNN with CIFAR-10 and CIFAR-100, SCRUB† reduces $\overline{\Delta}$ from 2.66 and 1.77 to 0.01, respectively. Similar improvements are observed across ViT and ResNet-18, confirming that URGE's favorable performance is not tied to a particular network architecture.

Moreover, Table D4 evaluates GA+URGE over all ten CIFAR-10 target classes. It achieves an average UA of 0.17 and an average $\overline{\Delta}$ of 1.07, suggesting that its effectiveness is generalisable across classes.

Table D2: Class unlearning results on AllCNN with CIFAR-10 (target class: 'automobile') and CIFAR-100 (target class: 'aquarium fish'). † indicates URGE-enhanced variants.

Class Unlearning								
Model	AllCNN, CIFAR-10				AllCNN, CIFAR-100			
Metric	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$
Original	$99.48_{\pm 0.00}$ (99.48)	$99.73_{\pm 0.00}$ (0.23)	$84.20_{\pm 0.00}$ (0.09)	33.27	$100.0_{\pm 0.00}$ (100.0)	$99.96_{\pm 0.00}$ (0.02)	$53.06_{\pm 0.00}$ (1.08)	33.70
Retrain	$0.00_{\pm 0.00}$ (0.00)	$99.96_{\pm 0.00}$ (0.00)	$84.29_{\pm 0.00}$ (0.00)	0.00	$0.00_{\pm 0.00}$ (0.00)	$99.98_{\pm 0.00}$ (0.00)	$54.14_{\pm 0.00}$ (0.00)	0.00
SatImp	$0.80_{\pm 0.08}$ (0.80)	$88.59_{\pm 0.95}$ (11.37)	$73.45_{\pm 1.15}$ (10.84)	7.67	$1.40_{\pm 0.15}$ (1.40)	$95.95_{\pm 0.20}$ (4.03)	$44.92_{\pm 1.50}$ (9.22)	4.88
WGA	$01.03_{\pm 0.12}$ (1.03)	$89.42_{\pm 0.90}$ (10.54)	$73.63_{\pm 1.10}$ (10.66)	7.41	$0.96_{\pm 0.11}$ (0.96)	$95.52_{\pm 0.25}$ (4.46)	$44.38_{\pm 1.55}$ (9.76)	5.06
LoReUn_s	$0.00_{\pm 0.00}$ (0.00)	$87.74_{\pm 1.05}$ (12.22)	$72.98_{\pm 1.25}$ (11.31)	7.84	$0.80_{\pm 0.10}$ (0.80)	$95.86_{\pm 0.22}$ (4.12)	$44.84_{\pm 1.45}$ (9.30)	4.74
LoReUn_d	$0.00_{\pm 0.00}$ (0.00)	$88.11_{\pm 1.00}$ (11.85)	$73.31_{\pm 1.20}$ (10.98)	7.61	$0.80_{\pm 0.10}$ (0.80)	$95.97_{\pm 0.18}$ (4.01)	$44.94_{\pm 1.40}$ (9.20)	4.67
GA	$0.26_{\pm 0.04}$ (0.26)	$88.44_{\pm 0.98}$ (11.52)	$73.20_{\pm 1.22}$ (11.09)	7.62	$1.00_{\pm 0.13}$ (1.00)	$79.03_{\pm 2.80}$ (20.95)	$44.59_{\pm 1.60}$ (9.55)	10.50
GA†	$0.02_{\pm 0.01}$ (0.02)	$99.17_{\pm 0.05}$ (0.79)	$82.83_{\pm 0.18}$ (1.46)	0.76	$0.08_{\pm 0.02}$ (0.08)	$98.74_{\pm 0.06}$ (1.24)	$47.13_{\pm 0.90}$ (7.01)	2.78
BS	$0.84_{\pm 0.10}$ (0.84)	$91.18_{\pm 0.75}$ (8.78)	$74.76_{\pm 1.05}$ (9.53)	6.38	$0.80_{\pm 0.10}$ (0.80)	$80.60_{\pm 2.10}$ (19.38)	$44.78_{\pm 1.65}$ (9.36)	9.85
BS†	$0.46_{\pm 0.06}$ (0.46)	$92.50_{\pm 0.65}$ (7.46)	$76.26_{\pm 0.90}$ (8.03)	5.32	$4.20_{\pm 1.30}$ (4.20)	$97.78_{\pm 0.08}$ (2.20)	$46.16_{\pm 1.20}$ (7.98)	4.79
SalUn	$0.40_{\pm 0.05}$ (0.40)	$89.31_{\pm 0.85}$ (10.65)	$75.63_{\pm 0.95}$ (8.66)	6.57	$0.34_{\pm 0.04}$ (0.34)	$80.27_{\pm 2.00}$ (19.71)	$44.86_{\pm 1.55}$ (9.28)	9.78
SalUn†	$0.78_{\pm 0.09}$ (0.78)	$92.01_{\pm 0.70}$ (7.95)	$76.89_{\pm 0.85}$ (7.40)	5.38	$0.10_{\pm 0.02}$ (0.10)	$99.86_{\pm 0.03}$ (0.12)	$49.00_{\pm 0.70}$ (5.14)	1.79
SCRUB	$0.00_{\pm 0.00}$ (0.00)	$97.20_{\pm 0.08}$ (2.76)	$79.08_{\pm 0.10}$ (5.21)	2.66	$0.00_{\pm 0.00}$ (0.00)	$96.98_{\pm 0.10}$ (3.00)	$51.83_{\pm 0.12}$ (2.31)	1.77
SCRUB†	$0.00_{\pm 0.00}$ (0.00)	$99.96_{\pm 0.00}$ (0.00)	$84.26_{\pm 0.00}$ (0.03)	**0.01**	$0.00_{\pm 0.00}$ (0.00)	$99.98_{\pm 0.00}$ (0.00)	$54.12_{\pm 0.02}$ (0.02)	**0.01**

Table D3: Class unlearning results on ViT with CIFAR-10 (target class: 'automobile'), and on ResNet-18 with CIFAR-100 (target class: 'aquarium fish'). † indicates URGE-enhanced variants.

Class Unlearning								
Model	ViT, CIFAR-10				ResNet-18, CIFAR-100			
Metric	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$
Original	$99.60_{\pm 0.00}(99.60)$	$99.28_{\pm 0.00}(0.64)$	$97.76_{\pm 0.00}(1.06)$	33.77	$100.0_{\pm 0.00}(100.0)$	$99.97_{\pm 0.00}(0.00)$	$52.43_{\pm 0.00}(0.35)$	33.45
Retrain	$0.00_{\pm 0.00}(0.00)$	$99.92_{\pm 0.00}(0.00)$	$98.82_{\pm 0.00}(0.00)$	0.00	$0.00_{\pm 0.00}(0.00)$	$99.97_{\pm 0.00}(0.00)$	$52.78_{\pm 0.00}(0.00)$	0.00
SatImp	$0.00_{\pm 0.00}(0.00)$	$97.99_{\pm 0.12}(1.93)$	$96.62_{\pm 0.18}(2.20)$	1.38	$1.80_{\pm 0.24}(1.80)$	$98.80_{\pm 0.08}(1.17)$	$43.41_{\pm 0.73}(9.37)$	4.11
WGA	$0.00_{\pm 0.00}(0.00)$	$94.20_{\pm 0.54}(5.72)$	$92.98_{\pm 0.48}(5.84)$	3.85	$1.21_{\pm 0.19}(1.21)$	$98.63_{\pm 0.09}(1.34)$	$43.26_{\pm 0.68}(9.52)$	4.02
LoReUn_s	$0.44_{\pm 0.07}(0.44)$	$98.07_{\pm 0.21}(1.85)$	$96.69_{\pm 0.15}(2.13)$	1.47	$1.40_{\pm 0.16}(1.40)$	$98.78_{\pm 0.11}(1.19)$	$43.38_{\pm 0.71}(9.40)$	4.00
LoReUn_d	$0.00_{\pm 0.00}(0.00)$	$89.17_{\pm 1.73}(10.75)$	$80.88_{\pm 2.15}(17.94)$	9.56	$1.60_{\pm 0.22}(1.60)$	$98.80_{\pm 0.13}(1.17)$	$43.42_{\pm 0.75}(9.36)$	4.04
GA	$0.32_{\pm 0.21}(0.32)$	$96.00_{\pm 0.31}(3.92)$	$94.58_{\pm 0.59}(4.24)$	2.83	$1.80_{\pm 0.31}(1.80)$	$93.87_{\pm 1.14}(6.10)$	$40.02_{\pm 1.62}(12.76)$	6.89
GA†	$0.00_{\pm 0.00}(0.00)$	$99.87_{\pm 0.01}(0.05)$	$98.49_{\pm 0.03}(0.33)$	**0.13**	$2.00_{\pm 0.44}(2.00)$	$99.18_{\pm 0.06}(0.79)$	$44.20_{\pm 0.58}(8.58)$	3.79
BS	$1.06_{\pm 0.01}(1.06)$	$83.60_{\pm 0.03}(16.32)$	$73.73_{\pm 2.67}(25.09)$	14.16	$1.36_{\pm 0.27}(1.36)$	$94.14_{\pm 0.89}(5.83)$	$40.98_{\pm 1.45}(11.80)$	6.33
BS†	$1.21_{\pm 0.36}(1.21)$	$86.77_{\pm 0.97}(13.15)$	$78.51_{\pm 1.53}(20.31)$	11.56	$1.80_{\pm 0.25}(1.80)$	$98.94_{\pm 0.07}(1.03)$	$43.59_{\pm 0.64}(9.19)$	4.01
SalUn	$11.76_{\pm 3.14}(11.76)$	$84.32_{\pm 1.68}(15.60)$	$75.22_{\pm 2.41}(23.60)$	16.99	$0.20_{\pm 0.04}(0.20)$	$95.51_{\pm 0.32}(4.46)$	$41.20_{\pm 0.98}(11.58)$	5.41
SalUn†	$7.82_{\pm 2.05}(7.82)$	$86.17_{\pm 1.23}(13.75)$	$78.35_{\pm 1.77}(20.47)$	14.01	$0.20_{\pm 0.02}(0.20)$	$99.96_{\pm 0.00}(0.01)$	$46.37_{\pm 0.02}(6.41)$	2.21
SCRUB	$0.00_{\pm 0.00}(0.00)$	$98.99_{\pm 0.01}(0.93)$	$97.82_{\pm 0.02}(1.00)$	0.64	$0.00_{\pm 0.00}(0.00)$	$99.97_{\pm 0.00}(0.00)$	$51.31_{\pm 0.01}(1.47)$	0.49
SCRUB†	$0.00_{\pm 0.00}(0.00)$	$99.60_{\pm 0.00}(0.32)$	$98.08_{\pm 0.01}(0.74)$	0.35	$0.00_{\pm 0.00}(0.00)$	$99.97_{\pm 0.00}(0.00)$	$52.53_{\pm 0.00}(0.25)$	**0.08**

Table D4: GA+URGE results with different target classes on AllCNN, CIFAR-10. † indicates URGE-enhanced variants.

Class Unlearning, AllCNN, CIFAR-10							
	Retrain			GA+URGE			
Class	UA ↓	RA ↑	TA ↑	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$
0	0.00	99.85	83.67	$0.80_{\pm 0.03}(0.80)$	$99.14_{\pm 0.06}(0.71)$	$82.44_{\pm 0.15}(1.23)$	0.91
1	0.00	99.96	84.29	$0.02_{\pm 0.01}(0.02)$	$99.17_{\pm 0.05}(0.79)$	$82.83_{\pm 0.18}(1.46)$	0.76
2	0.00	99.96	84.48	$0.03_{\pm 0.01}(0.03)$	$98.71_{\pm 0.07}(1.25)$	$82.44_{\pm 0.22}(2.04)$	1.11
3	0.00	99.98	85.02	$0.06_{\pm 0.02}(0.06)$	$98.97_{\pm 0.04}(1.01)$	$83.97_{\pm 0.13}(1.05)$	0.71
4	0.00	99.96	84.98	$0.13_{\pm 0.04}(0.13)$	$98.46_{\pm 0.08}(1.50)$	$82.26_{\pm 0.25}(2.72)$	1.45
5	0.00	99.88	84.57	$0.14_{\pm 0.03}(0.14)$	$98.47_{\pm 0.05}(1.41)$	$82.11_{\pm 0.20}(2.46)$	1.34
6	0.00	99.90	84.13	$0.07_{\pm 0.02}(0.07)$	$99.07_{\pm 0.06}(0.83)$	$82.17_{\pm 0.18}(1.96)$	0.95
7	0.00	99.90	84.38	$0.02_{\pm 0.01}(0.02)$	$98.50_{\pm 0.07}(1.40)$	$81.21_{\pm 0.24}(3.17)$	1.53
8	0.00	99.94	84.25	$0.38_{\pm 0.05}(0.38)$	$99.09_{\pm 0.04}(0.85)$	$83.44_{\pm 0.12}(0.81)$	0.68
9	0.00	99.89	84.40	$0.04_{\pm 0.02}(0.04)$	$98.51_{\pm 0.05}(1.38)$	$81.93_{\pm 0.21}(2.47)$	1.30
Avg.	0.00	99.92	84.42	$0.17_{\pm 0.25}(0.17)$	$98.81_{\pm 0.31}(1.11)$	$82.48_{\pm 0.78}(1.94)$	1.07

Random Unlearning Results

Across Tables D5–D7, URGE-enhanced GU generally yield smaller retrain-oracle gaps than their corresponding base methods, with SCRUB† providing the most close approximation to the retraining oracle across architectures and datasets.

For random unlearning, Tables D5–D7 show that URGE generally improves the forgetting–utility trade-off of the base GU methods across different architectures and datasets. GA, BS, and SalUn remain relatively far from the retraining oracle because they do not use retain data or explicit retain regularization, which is particularly important when the randomly selected forget and retain samples follow the same distribution. This limitation originates from the base GU objectives rather than URGE. Notably, when combined with the retain-regularized SCRUB, URGE generally achieves the smallest $\overline{\Delta}$ across all six settings, demonstrating the effectiveness of its fine-grained forget-side reweighting.

Table D5: Random unlearning results on ResNet18 with CIFAR-10 (forget ratio=10%) and CIFAR-100 (forget ratio=1%).† indicates URGE-enhanced variants.

Random Unlearning								
Model	ResNet-18, CIFAR-10, 10%				ResNet-18, CIFAR-100, 1%			
Metric	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$
Original	$99.86_{\pm 0.00}$(16.32)	$99.84_{\pm 0.00}$(0.08)	$92.50_{\pm 0.00}$(0.67)	5.69	$100.0_{\pm 0.00}$(35.86)	$99.98_{\pm 0.00}$(0.00)	$52.43_{\pm 0.00}$(0.49)	12.12
Retrain	$83.54_{\pm 0.00}$(0.00)	$99.92_{\pm 0.00}$(0.00)	$93.17_{\pm 0.00}$(0.00)	0.00	$64.14_{\pm 0.00}$(0.00)	$99.98_{\pm 0.00}$(0.00)	$52.92_{\pm 0.00}$(0.00)	0.00
SatImp	$98.52_{\pm 0.44}$(14.98)	$98.39_{\pm 0.27}$(1.53)	$90.89_{\pm 0.31}$(2.28)	6.26	$97.13_{\pm 0.42}$(32.99)	$97.10_{\pm 0.30}$(2.88)	$41.89_{\pm 0.73}$(11.03)	15.63
WGA	$92.52_{\pm 0.26}$(8.98)	$93.17_{\pm 0.48}$(6.75)	$86.07_{\pm 0.59}$(7.10)	7.61	$96.62_{\pm 0.39}$(32.48)	$96.89_{\pm 0.33}$(3.09)	$42.37_{\pm 0.68}$(10.55)	15.37
LoReUn_s	$92.40_{\pm 0.24}$(8.86)	$93.16_{\pm 0.51}$(6.76)	$86.14_{\pm 0.56}$(7.03)	7.55	$96.78_{\pm 0.41}$(32.64)	$97.01_{\pm 0.31}$(2.97)	$42.11_{\pm 0.71}$(10.81)	15.47
LoReUn_d	$98.92_{\pm 0.47}$(15.38)	$98.81_{\pm 0.21}$(1.11)	$91.36_{\pm 0.27}$(1.81)	6.10	$97.03_{\pm 0.43}$(32.89)	$97.38_{\pm 0.28}$(2.60)	$41.66_{\pm 0.75}$(11.26)	15.58
GA	$93.22_{\pm 0.29}$(9.68)	$93.99_{\pm 0.43}$(5.93)	$86.76_{\pm 0.53}$(6.41)	7.34	$96.84_{\pm 0.40}$(32.70)	$97.12_{\pm 0.29}$(2.86)	$41.95_{\pm 0.70}$(10.97)	15.51
GA†	$95.34_{\pm 0.33}$(11.80)	$96.28_{\pm 0.34}$(3.64)	$88.41_{\pm 0.45}$(4.76)	6.73	$96.49_{\pm 0.37}$(32.35)	$96.98_{\pm 0.32}$(3.00)	$41.90_{\pm 0.74}$(11.02)	15.46
BS	$98.90_{\pm 0.46}$(15.36)	$99.05_{\pm 0.19}$(0.87)	$91.70_{\pm 0.23}$(1.47)	5.90	$68.74_{\pm 0.15}$(4.60)	$97.40_{\pm 0.26}$(2.58)	$42.40_{\pm 0.66}$(10.52)	5.90
BS†	$91.92_{\pm 0.22}$(8.38)	$95.85_{\pm 0.54}$(4.07)	$88.58_{\pm 0.62}$(4.59)	5.68	$64.88_{\pm 0.08}$(0.74)	$95.92_{\pm 0.39}$(4.06)	$41.76_{\pm 0.76}$(11.16)	5.32
SalUn	$93.56_{\pm 0.31}$(10.02)	$93.18_{\pm 0.49}$(6.74)	$85.16_{\pm 0.66}$(8.01)	8.26	$97.43_{\pm 0.45}$(33.29)	$98.81_{\pm 0.20}$(1.17)	$42.68_{\pm 0.64}$(10.24)	14.90
SalUn†	$98.88_{\pm 0.45}$(15.34)	$99.15_{\pm 0.17}$(0.77)	$91.41_{\pm 0.28}$(1.76)	5.96	$95.03_{\pm 0.34}$(30.89)	$97.69_{\pm 0.25}$(2.29)	$41.90_{\pm 0.72}$(11.02)	14.73
SCRUB	$90.76_{\pm 0.19}$(7.22)	$97.14_{\pm 0.30}$(2.78)	$87.38_{\pm 0.50}$(5.79)	5.26	$64.43_{\pm 0.02}$(0.29)	$99.96_{\pm 0.00}$(0.02)	$50.98_{\pm 0.02}$(1.94)	0.75
SCRUB†	$93.96_{\pm 0.30}$(10.42)	$99.22_{\pm 0.15}$(0.70)	$88.89_{\pm 0.43}$(4.28)	**5.13**	$64.41_{\pm 0.03}$(0.27)	$99.96_{\pm 0.00}$(0.02)	$51.02_{\pm 0.02}$(1.90)	**0.73**

Table D6: Random unlearning results on AllCNN with CIFAR-10 (forget ratio=10%) and CIFAR-100 (forget ratio=1%). † indicates URGE-enhanced variants.

Random Unlearning								
Model	AllCNN, CIFAR-10, 10%				AllCNN, CIFAR-100, 1%			
Metric	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$
Original	$99.72_{\pm 0.53}$(15.78)	$99.68_{\pm 0.10}$(0.03)	$84.20_{\pm 0.28}$(0.96)	5.59	$100.0_{\pm 0.58}$(39.00)	$99.96_{\pm 0.08}$(0.02)	$53.06_{\pm 0.34}$(0.05)	13.02
Retrain	$83.94_{\pm 0.16}$(0.00)	$99.71_{\pm 0.09}$(0.00)	$83.24_{\pm 0.31}$(0.00)	0.00	$61.00_{\pm 0.14}$(0.00)	$99.98_{\pm 0.06}$(0.00)	$53.11_{\pm 0.32}$(0.00)	0.00
SatImp	$99.63_{\pm 0.02}$(15.69)	$99.51_{\pm 0.04}$(0.20)	$82.86_{\pm 0.07}$(0.38)	5.42	$98.97_{\pm 0.46}$(37.97)	$99.04_{\pm 0.17}$(0.94)	$51.02_{\pm 0.41}$(2.09)	13.67
WGA	$99.10_{\pm 0.14}$(15.16)	$99.37_{\pm 0.16}$(0.34)	$82.32_{\pm 0.43}$(0.92)	5.47	$96.20_{\pm 0.38}$(35.20)	$97.06_{\pm 0.31}$(2.92)	$48.22_{\pm 0.56}$(4.89)	14.34
LoReUn_s	$99.25_{\pm 0.07}$(15.31)	$98.26_{\pm 0.17}$(1.45)	$82.74_{\pm 0.29}$(0.50)	5.75	$99.13_{\pm 0.51}$(38.13)	$99.19_{\pm 0.15}$(0.79)	$50.75_{\pm 0.44}$(2.36)	13.76
LoReUn_d	$98.70_{\pm 0.42}$(14.76)	$98.99_{\pm 0.22}$(0.72)	$82.06_{\pm 0.47}$(1.18)	5.55	$98.60_{\pm 0.45}$(37.60)	$98.96_{\pm 0.19}$(1.02)	$50.38_{\pm 0.48}$(2.73)	13.78
GA	$99.12_{\pm 0.25}$(15.18)	$99.32_{\pm 0.13}$(0.39)	$82.65_{\pm 0.41}$(0.59)	5.39	$99.37_{\pm 0.05}$(38.37)	$99.75_{\pm 0.09}$(0.23)	$50.82_{\pm 0.13}$(2.29)	13.63
GA†	$99.01_{\pm 0.13}$(15.07)	$99.45_{\pm 0.08}$(0.26)	$82.79_{\pm 0.16}$(0.45)	5.26	$97.20_{\pm 0.04}$(36.20)	$99.83_{\pm 0.07}$(0.15)	$52.62_{\pm 0.05}$(0.49)	12.28
BS	$98.48_{\pm 0.36}$(14.54)	$99.10_{\pm 0.18}$(0.61)	$82.42_{\pm 0.44}$(0.82)	5.32	$69.20_{\pm 0.23}$(8.20)	$89.78_{\pm 0.73}$(10.20)	$37.50_{\pm 0.91}$(15.61)	11.34
BS†	$98.01_{\pm 0.34}$(14.07)	$98.98_{\pm 0.23}$(0.73)	$82.31_{\pm 0.46}$(0.93)	5.24	$67.60_{\pm 0.21}$(6.60)	$90.44_{\pm 0.68}$(9.54)	$38.11_{\pm 0.87}$(15.00)	10.38
SalUn	$98.34_{\pm 0.35}$(14.40)	$98.17_{\pm 0.31}$(1.54)	$81.45_{\pm 0.54}$(1.79)	5.91	$95.20_{\pm 0.37}$(34.20)	$95.69_{\pm 0.46}$(4.29)	$41.72_{\pm 0.72}$(11.39)	16.63
SalUn†	$99.04_{\pm 0.46}$(15.10)	$99.16_{\pm 0.17}$(0.55)	$82.44_{\pm 0.42}$(0.80)	5.48	$96.21_{\pm 0.39}$(35.21)	$96.95_{\pm 0.35}$(3.03)	$43.29_{\pm 0.65}$(9.82)	16.02
SCRUB	$85.45_{\pm 0.18}$(1.51)	$99.38_{\pm 0.01}$(0.33)	$83.02_{\pm 0.03}$(0.22)	0.69	$66.32_{\pm 0.05}$(5.32)	$97.86_{\pm 0.04}$(2.12)	$48.87_{\pm 0.08}$(4.24)	3.89
SCRUB†	$84.98_{\pm 0.02}$(1.04)	$99.12_{\pm 0.04}$(0.59)	$82.85_{\pm 0.08}$(0.39)	**0.67**	$66.56_{\pm 0.02}$(5.56)	$98.61_{\pm 0.02}$(1.37)	$49.73_{\pm 0.07}$(3.38)	**3.44**

Table D7: Random unlearning results on ViT with CIFAR-10 (forget ratio=10%) and CIFAR-100 (forget ratio=1%). † indicates URGE-enhanced variants.

Random Unlearning								
Model	ViT, CIFAR-10, 10%				ViT, CIFAR-100, 1%			
Metric	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$
Original	$99.35_{\pm 0.00}$(6.61)	$99.32_{\pm 0.00}$(0.33)	$97.76_{\pm 0.00}$(0.06)	2.33	$98.10_{\pm 0.00}$(2.60)	$97.68_{\pm 0.00}$(0.53)	$89.16_{\pm 0.00}$(0.08)	1.07
Retrain	$92.74_{\pm 0.00}$(0.00)	$99.65_{\pm 0.00}$(0.00)	$97.70_{\pm 0.00}$(0.00)	0.00	$95.50_{\pm 0.00}$(0.00)	$98.21_{\pm 0.00}$(0.00)	$89.08_{\pm 0.00}$(0.00)	0.00
SatImp	$98.52_{\pm 0.35}$(5.78)	$98.59_{\pm 0.29}$(1.06)	$96.10_{\pm 0.38}$(1.60)	2.81	$97.87_{\pm 0.32}$(2.37)	$98.01_{\pm 0.21}$(0.20)	$88.95_{\pm 0.29}$(0.13)	0.90
WGA	$97.29_{\pm 0.28}$(4.55)	$97.27_{\pm 0.37}$(2.38)	$95.04_{\pm 0.46}$(2.66)	3.20	$95.83_{\pm 0.20}$(0.33)	$96.35_{\pm 0.42}$(1.86)	$87.02_{\pm 0.48}$(2.06)	1.42
LoReUn_s	$98.01_{\pm 0.31}$(5.27)	$98.33_{\pm 0.33}$(1.32)	$95.95_{\pm 0.40}$(1.75)	2.78	$96.89_{\pm 0.25}$(1.39)	$97.25_{\pm 0.35}$(0.96)	$87.58_{\pm 0.43}$(1.50)	1.28
LoReUn_d	$98.69_{\pm 0.39}$(5.95)	$98.76_{\pm 0.25}$(0.89)	$97.03_{\pm 0.32}$(0.67)	2.50	$97.33_{\pm 0.30}$(1.83)	$97.48_{\pm 0.31}$(0.73)	$87.44_{\pm 0.45}$(1.64)	1.40
GA	$98.74_{\pm 0.43}$(6.00)	$98.94_{\pm 0.22}$(0.71)	$96.79_{\pm 0.36}$(0.91)	2.54	$96.40_{\pm 0.23}$(0.90)	$97.04_{\pm 0.38}$(1.17)	$87.83_{\pm 0.39}$(1.25)	1.11
GA†	$99.01_{\pm 0.02}$(6.27)	$99.56_{\pm 0.01}$(0.09)	$97.13_{\pm 0.04}$(0.57)	2.31	$95.63_{\pm 0.19}$(0.13)	$97.11_{\pm 0.36}$(1.10)	$87.86_{\pm 0.37}$(1.22)	0.82
BS	$98.88_{\pm 0.45}$(6.14)	$99.09_{\pm 0.19}$(0.56)	$96.92_{\pm 0.34}$(0.78)	2.49	$97.56_{\pm 0.33}$(2.06)	$97.89_{\pm 0.22}$(0.32)	$88.01_{\pm 0.34}$(1.07)	1.15
BS†	$98.98_{\pm 0.46}$(6.24)	$99.14_{\pm 0.17}$(0.51)	$97.00_{\pm 0.30}$(0.70)	2.48	$97.10_{\pm 0.27}$(1.60)	$97.66_{\pm 0.28}$(0.55)	$87.98_{\pm 0.36}$(1.10)	1.08
SalUn	$98.39_{\pm 0.34}$(5.65)	$98.54_{\pm 0.30}$(1.11)	$95.99_{\pm 0.42}$(1.71)	2.82	$97.98_{\pm 0.35}$(2.48)	$98.13_{\pm 0.18}$(0.08)	$87.69_{\pm 0.41}$(1.39)	1.32
SalUn†	$99.09_{\pm 0.49}$(6.35)	$99.14_{\pm 0.16}$(0.51)	$97.02_{\pm 0.31}$(0.68)	2.51	$97.43_{\pm 0.29}$(1.93)	$97.80_{\pm 0.26}$(0.41)	$88.34_{\pm 0.33}$(0.74)	1.03
SCRUB	$92.86_{\pm 0.14}$(0.12)	$99.28_{\pm 0.15}$(0.37)	$97.15_{\pm 0.28}$(0.55)	0.35	$95.98_{\pm 0.21}$(0.48)	$97.89_{\pm 0.23}$(0.32)	$88.74_{\pm 0.27}$(0.34)	0.38
SCRUB†	$92.90_{\pm 0.03}$(0.16)	$99.49_{\pm 0.01}$(0.16)	$97.41_{\pm 0.04}$(0.29)	**0.20**	$96.03_{\pm 0.22}$(0.53)	$98.11_{\pm 0.07}$(0.10)	$88.92_{\pm 0.25}$(0.16)	**0.26**

SEQUENTIAL UNLEARNING RESULTS

The sequential unlearning results further confirm the applicability of URGE beyond one-shot class removal. In the sequential setting of Table D8, repeated deletion requests amplify the risk of utility degradation because each unlearning stage operates on an already modified model. URGE nevertheless generally improves the balance between removing the accumulated target classes and preserving the remaining knowledge. In particular, SCRUB† consistently reduces UA across all four stages and usually improves RA and TA over SCRUB, demonstrating that the representation-guided dynamic weighting remains effective under accumulated model updates.

Table D8: Sequential class unlearning results on ResNet-18, CIFAR-10. † indicates URGE-enhanced variants.

Sequential Unlearning																
Model	**Class 1+7**				**Class 1+7+3**				**Class 1+7+3+9**				**Class 1+7+3+9+0**			
Metric	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$
Retrain	0.00	99.48	86.48	0.00	0.00	99.44	82.57	0.00	0.00	99.57	83.22	0.00	0.00	99.88	81.68	0.00
SatImp	5.21	96.73	80.18	4.75	5.07	88.42	75.05	7.87	4.16	77.18	66.58	14.40	6.12	86.34	73.88	9.15
WGA	5.84	96.15	79.67	5.33	5.53	87.90	74.60	8.35	4.68	76.50	65.90	15.02	6.54	85.70	73.10	9.77
$\mathrm{LoReUn_s}$	2.29	90.36	74.59	7.77	4.99	89.93	77.96	6.37	3.42	80.06	68.42	12.58	6.09	82.38	69.94	11.78
$\mathrm{LoReUn_d}$	4.37	97.22	80.89	4.07	4.89	88.76	75.43	7.57	3.95	77.85	67.12	13.92	5.73	87.05	74.28	8.65
GA	4.51	97.11	80.26	4.37	2.09	83.60	75.49	8.34	1.92	77.98	66.98	13.25	3.23	85.38	73.71	8.57
GA†	1.63	99.51	83.02	1.71	2.13	91.52	77.86	4.92	1.04	80.04	69.48	11.44	2.89	89.45	77.04	5.99
BS	4.76	99.02	81.23	3.49	2.41	94.81	79.77	3.28	4.15	88.55	76.72	7.22	1.02	84.07	72.90	8.54
BS†	2.93	99.11	81.64	2.71	2.55	95.49	80.64	2.81	2.84	90.66	77.88	5.70	1.45	87.34	76.01	6.55
SalUn	2.51	94.93	80.12	4.47	1.19	83.14	73.84	8.07	2.32	85.02	72.65	9.15	2.98	86.18	74.32	8.01
SalUn†	1.23	98.77	82.12	2.10	0.49	84.98	73.86	7.89	1.05	84.02	73.08	8.91	0.48	84.98	73.53	7.85
SCRUB	1.08	99.01	83.27	1.59	0.53	91.87	78.99	3.89	0.82	93.95	77.85	3.94	0.97	89.76	78.13	4.88
SCRUB†	0.27	98.86	82.72	1.55	0.05	92.03	80.37	3.22	0.69	95.29	78.95	3.08	0.31	90.80	79.22	3.95

MIA RESULTS

We evaluate the privacy protection of different models against membership inference attacks (MIA). Following the basic MIA protocol[1], we use 60% of the training forget data to fit an SVC attacker and evaluate its attack accuracy on the remaining 40%. Specifically, we compare the attack success rate (ASR) of the original model, the retrained model, baseline-unlearned models, and our URGE-unlearned model. A lower ASR indicates that the model exposes less membership information and is therefore more resistant to privacy leakage. As shown in Table D9, the original model is more vulnerable to MIA, while the retrained model provides a reference for ideal unlearning. Compared with baselines, our method achieves a lower ASR and is closer to the retrained model, suggesting that URGE more effectively removes membership information associated with the forgotten data.

[1]Shokri, Reza, et al. "Membership inference attacks against machine learning models." 2017 IEEE symposium on security and privacy (SP). IEEE, 2017.

Table D9: MIA attach successful ratio (ASR) results for image classification task. † indicates URGE-enhanced variants.

Dataset	CIFAR-10			CIFAR-100		
Method	AllCNN	ResNet-18	ViT	AllCNN	ResNet-18	ViT
Original	98.00	100.0	100.0	100.0	100.0	100.0
Retrain	0.00	0.00	0.00	0.00	0.00	0.00
GA	2.75	2.50	0.00	2.75	5.75	0.00
GA†	0.00	0.00	0.00	0.00	0.00	0.00
BS	1.50	1.00	0.50	2.50	1.75	0.00
BS†	0.00	0.00	0.25	0.00	0.50	0.00
SalUn	0.00	0.00	10.25	2.50	0.00	0.00
SalUn†	0.00	0.00	6.50	0.00	0.00	0.00
SCRUB	0.00	0.00	0.00	0.00	0.00	0.00
SCRUB†	0.00	0.00	0.00	0.00	0.00	0.00

D.2 Extended Results for LLM Unlearning

LLM Unlearning on WMDP.

We further evaluate URGE on WMDP using Zephyr-7B, where lower unlearning accuracy indicates stronger removal of hazardous knowledge and higher MMLU accuracy reflects better preservation of general capabilities. As shown in Table D10, incorporating URGE generally improves the forgetting–utility trade-off of vanilla GU baselines across the biological and cybersecurity domains. In particular, URGE substantially reduces unlearning accuracy while largely preserving or even improving MMLU accuracy. These results demonstrate that representation-guided reweighting promotes more targeted removal of hazardous knowledge without causing excessive degradation of general utility.

Table D10: Comparison of baselines on WMDP. † indicates URGE-enhanced variants.

Method	**WMDP-bio**		**WMDP-cyber**	
	Unlearn Acc ↓	MMLU Acc ↑	Unlearn Acc ↓	MMLU Acc ↑
Original	0.6442	0.5853	0.4432	0.5849
SatImp	0.3677	0.4594	0.3684	0.5373
WGA	0.3590	0.4535	0.3544	0.5357
WTNPO	0.3669	0.4623	0.3625	0.5341
GA	0.4024	0.4694	0.3923	0.5321
GA†	0.3515	0.4647	0.3151	0.5338
NPO	0.3401	0.4942	0.3439	0.5598
NPO†	0.3052	0.5144	0.3203	0.5569
RMU	0.3570	0.4886	0.3708	0.5395
RMU†	0.2812	0.5018	0.3496	0.5424

LLM Unlearning on MUSE.

We further evaluate LLM unlearning performance on the MUSE-books benchmark with the Llama2-7B-chat model. As shown in Table D11, compared with the GU baseline, URGE achieves a significantly stronger removal of forgetting knowledge while preserving comparable retain utility on KnowMem $\mathcal{D}_r(\uparrow)$.

Table D11: Comparison of baselines on MUSE. † indicates URGE-enhanced variants.

Method	Unlearning Efficacy			Utility
	VerbMem $\mathcal{D}_f$ ↓)	KnowMem $\mathcal{D}_f$(↓)	PrivLeak (→ 0)	KnowMem $\mathcal{D}_r$(↑)
Original	0.9970	0.4712	-57.3410	0.6913
Retrain	0.1445	0.3029	8.1600	0.6874
SatImp	0.1694	0.2004	-46.0527	0.3095
WGA	0.2093	0.2844	-46.5323	0.3929
WTNPO	0.2015	0.2702	-46.1536	0.4069
GA	0.5226	0.3272	-47.2397	0.5582
GA†	0.2428	0.2284	-51.9822	0.5109
NPO	0.3327	0.2371	-41.0152	0.5390
NPO†	0.1752	0.2910	-42.4972	0.5542
RMU	0.1634	0.3437	-41.3116	0.6110
RMU†	0.1446	0.3709	-39.4224	0.6819

D.3 Extended Results for Ablation Study

Module-level Ablation

In this section, we further provide extended module ablation results across different datasets and models in the image classification task. Consistent with the observations in the main paper, the results in Figure D13 and D14 demonstrate the advantages of incorporating CRI and RS for improving unlearning performance. Individually, both CRI and RS can provide noticeable performance gains, while their combination achieves an even better forgetting–utility trade-off, approaching the performance of the retrain-from-scratch oracle.

Table D12: Ablation study of CRI and RS on the unlearning of image classification and diffusion models.

CRI	RS	**Image Classification** *GA-RN18-CIFAR10*				**DMs** *RL-DDPM-CIFAR10*	
		UA ↓	RA ↑	TA ↑	$\overline{\Delta}$ ↓	UA ↓	FID ↓
-	-	4.86	85.82	79.77	10.34	0.00	22.82
✓	-	6.86	91.22	84.19	7.73	0.00	21.82
-	✓	1.12	91.68	84.21	5.66	0.00	21.49
✓	✓	0.91	92.40	84.43	**5.27**	0.00	**21.30**

Table D13: Ablation study of CRI and RS on image classification with AllCNN and ResNet-18 model. † indicates URGE-enhanced variants.

CRI	RS	**AllCNN, GA** *CIFAR-10*				**AllCNN, GA** *CIFAR-100*				**ResNet-18, GA** *CIFAR-100*			
		UA ↓	RA ↑	TA ↑	$\overline{\Delta}$ ↓	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$ ↓	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$ ↓
-	-	0.26	88.44	73.20	7.62	1.00	79.03	44.59	10.50	1.80	93.87	40.02	6.89
✓	-	0.42	90.83	76.05	5.93	3.20	93.78	42.17	7.12	1.20	98.73	43.24	3.99
-	✓	0.04	98.84	82.86	0.86	4.00	97.85	46.18	4.70	1.60	98.85	43.53	3.99
✓	✓	0.02	99.17	82.83	0.76	0.08	98.74	47.13	2.78	2.00	99.18	44.20	3.79

Table D14: Ablation study of CRI and RS on image classification with ViT model.

CRI	RS	ViT, GA *CIFAR-10*				ViT, GA *CIFAR-100*			
		UA ↓	RA ↑	TA ↑	$\overline{\Delta}$ ↓	UA ↓	RA ↑	TA ↑	$\overline{\Delta}$ ↓
-	-	0.32	96.00	94.58	2.83	0.00	95.62	87.66	3.09
✓	-	0.30	98.00	96.58	1.49	0.00	97.68	89.59	1.29
-	✓	0.00	98.85	97.46	0.81	0.00	96.76	88.77	1.87
✓	✓	0.00	99.87	98.49	**0.13**	0.00	98.33	90.04	**1.39**

ABLATION ON PERTURBATION CHOICES FOR THE FORMING CONDITION.

We investigate how the choice of forming condition and its perturbation affects CRI. In token-level LLM unlearning, for the perturbation of the question condition, we compare our default semantic perturbation with Gaussian perturbation at different noise scales. AUC measures how well the resulting CRI scores distinguish critical from non-critical tokens according to the LLM-as-Judge annotations.

As shown in Tables D15 and D16, semantic perturbation achieves a highest AUC of 0.8250, substantially outperforming all parameter-noise variants, whose AUC values remain below 0.49. It also produces markedly higher FQ for both GA+CRI and the complete GA+URGE, although with some reduction in retain-side metrics due to stronger forgetting. In particular, GA+URGE with semantic perturbation achieves a highest FQ of 0.99. These results indicate that the question is a more appropriate forming condition for token-level CRI because it directly controls the semantics that elicit the answer tokens, whereas global parameter noise provides a weaker and less discriminative criticality signal. The results across different noise scales further suggest that the advantage comes from the choice of forming condition rather than sensitive tuning of the perturbation magnitude.

Table D15: Ablation on the Perturbation Choices of the Forming-Condition, on Llama3.2-1B, TOFU-Forget1%, GA+URGE.

perturbation	AUC ↑	ES-u ↓	ES-r ↑	MU ↑	FQ ↑
noise 10^{-5}	0.4451	0.0746	0.3539	0.5576	0.5786
noise 10^{-4}	0.4534	0.0820	0.3967	0.5590	0.4046
noise 10^{-3}	0.4884	0.1005	0.4084	0.5599	0.7659
noise 10^{-2}	0.4862	0.1198	0.3610	0.5574	0.1650
semantic	0.8250	0.0537	0.2359	0.5410	**0.9900**

Table D16: Ablation on the Perturbation Choises of the Forming-Condition, on Llama3.2-1B, TOFU-Forget1%, GA+CRI.

perturbation	AUC ↑	ES-u ↓	ES-r ↑	MU ↑	FQ ↑
noise 10^{-5}	0.4553	0.1210	0.5407	0.5836	0.0541
noise 10^{-4}	0.4534	0.1150	0.5411	0.5846	0.0541
noise 10^{-3}	0.4884	0.1199	0.5366	0.5838	0.0541
noise 10^{-2}	0.4862	0.1192	0.5413	0.5823	0.0971
semantic	0.8250	0.0922	0.3411	0.5570	**0.4046**

ABLATION ON RS STRATEGY SELECTION.

We examine whether RS should be recomputed dynamically throughout training and whether the focal threshold τ is necessary. We conduct this ablation on Llama3.2-1B with TOFU-Forget1% using GA+RS, while keeping all optimization hyperparameters fixed. *Dynamic* RS recomputes the

token-wise representation similarity at every training update. *static* RS computes these similarities only once after the first parameter update, and reuses the resulting scores throughout the remaining training process. For each strategy, we compare scoring with and without the threshold τ.

As shown in Table D17, the default dynamic RS strategy with τ achieves the best overall trade-off. In contrast, removing τ markedly degrades FQ, regardless of whether RS is computed dynamically or kept static, indicating that thresholding is critical for suppressing fast forgotten tokens. Static RS further reduces retention performance, suggesting that updating similarity scores during training better tracks the evolving model state. Although dynamic RS without τ attains a marginally higher MU, its much lower FQ confirms that dynamic recomputation alone is insufficient; both dynamic scoring and thresholded focal weighting are required for effective fine-grained unlearning.

Table D17: Ablation on RS with/without τ and dynamic vs. static computation, on Llama3.2-1B, TOFU-Forget1%, GA+RS.

Settings	ES-u ↓	ES-r ↑	MU ↑	FQ ↑
static, without τ	0.0907	0.2590	0.5396	0.2657
static, with τ	0.0845	0.2466	0.5343	0.2657
dynamic, without τ	0.0838	0.2748	0.5454	0.2657
dynamic, with τ (default)	0.0754	0.3014	0.5446	**0.7659**

Ablation on RS similarity functions.

Furthermore, we investigate alternative signals for constructing RS weights in GA+RS. The first two settings operate in the output space: *probability change* measures the change in each token's predicted probability relative to the original model, whereas *logits KL divergence* measures the discrepancy between the current and reference token-level predictive distributions.

As shown in Table D18, output-space discrepancies and representation distances either lead to limited forgetting quality or substantially degrade retention. Pearson correlation offers a stronger balance, but cosine similarity achieves the best overall result, attaining the highest FQ (0.7659) while maintaining competitive ES-u, ES-r, and MU. This suggests that angular alignment between current and reference representations provides a more reliable signal for identifying tokens that should be emphasized during unlearning than probability-level changes or magnitude-sensitive representation distances.

Table D18: Ablation on RS with varying representation similarity functions, on Llama3.2-1B, TOFU-Forget1%, GA+RS.

Similarity Function	ES-u ↓	ES-r	MU ↑	FQ ↑
Probability change	0.0441	0.2499	0.5482	0.0541
Logits KL divergence	0.0673	0.2905	0.5156	0.0143
Representation L1 distance	0.1683	0.5191	0.5778	0.0030
Representation L2 distance	0.1650	0.5186	0.5775	0.0030
Representation Pearson correlation	0.0892	0.4343	0.5725	0.4046
Representation Cosine similarity	0.0754	0.3014	0.5446	**0.7659**

Ablation on binary/continuous weighting.

This ablation evaluates binary and continuous token weighting on Llama-3.2-1B using TOFU-Forget1% with GA+URGE (token-level), with results shown in Table D19. For binary weighting, the continuous URGE weights are first computed, after which only the top-K fraction of valid forget-set tokens is assigned weight 1 and all remaining tokens are assigned weight 0. Overall, hard filtering shows unstable trade-offs as K varies, whereas continuous weighting provides a better balance between forgetting and utility. Meanwhile, a small K generally provides less retain utility preservation (with $K = 5\%$ being an exception), indicating that hard-filtering fewer tokens makes it

difficult to achieve fine-grained unlearning control.This suggests that retaining graded token importance is preferable to discarding it through binary filtering.

Table D19: Ablation on binary reweighting with varying filtering ratio, on Llama3.2-1B, TOFU-Forget1%, GA+URGE.

setting	ratio K	ES-u ↓	ES-r↑	MU ↑	FQ ↑
binary (hard filtering)	5%	0.0479	0.1468	0.5098	0.4046
	10%	0.0363	0.0685	0.3141	0.1650
	20%	0.0376	0.0910	0.3894	0.7659
	30%	0.0916	0.3613	0.5611	0.1650
	40%	0.0976	0.4238	0.5723	0.4046
	50%	0.0911	0.3521	0.5645	0.9188
continuous(URGE)	-	0.0537	0.2359	0.5410	0.9900

D.4 Hyperparameter Analysis

We conduct a hyperparameter sensitivity analysis on Llama3.2-1B with TOFU-forget01%, as well as on AllCNN with CIFAR-10. We evaluate the Representation Similarity threshold τ, the focal exponent γ, the CRI scaling bounds $\lambda_{\min}$ and $\lambda_{\max}$, and the representation extraction layer. For the latter, we consider two distinct settings: one varies only the representation layer used by CRI, while the other varies the extraction layer of both CRI and RS. We report FQ and ES-u for forgetting efficacy, and MU and ES-r for utility preservation.

Representation Similarity Threshold τ. As shown in Figure D1, τ directly controls when RS stops emphasizing a token. A small threshold permits continued updates after substantial representation shifts and may lead to over-unlearning, whereas an excessively large threshold can terminate the updates prematurely. A moderately high threshold provides the best balance, with $\tau = 0.88$ achieving an FQ of 0.99 while maintaining competitive MU and ES-r. A similar trend is observed for AllCNN on CIFAR-10 in Figure D7: intermediate thresholds yield low UA and a small $\overline{\Delta}$, while overly large values noticeably degrade both forgetting and retained accuracy. This confirms the importance of the token-level early-stopping behavior introduced by RS.

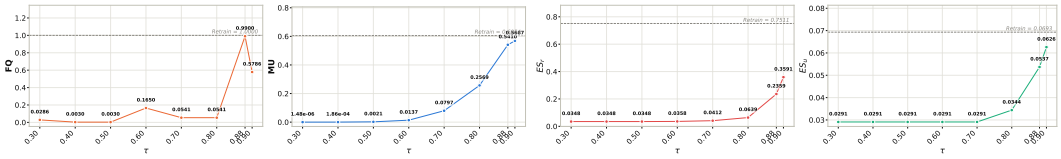
Representation Layer for CRI Only. Figure D2 isolates the layer sensitivity of CRI by varying only the representation used for CRI extraction in GA+token-level CRI. Here, 'logits' denotes a comparison setting in which the model's final output logits, rather than an intermediate hidden representation, are used to compute CRI. Later hidden layers generally provide a better forgetting–utility trade-off than early layers, with layer 15 achieving the strongest overall performance. In contrast, the logits-based variant performs notably worse, indicating that internal representations retain richer information for estimating the intrinsic forgetting need than the final predictive output.

Shared Representation Layer for Both CRI and RS. Figure D3 jointly varies the representation used by both CRI and RS in the complete GA+URGE framework. Accordingly, 'logits' means that the final output logits replace hidden representations in computing both the static CRI score and the dynamic RS score. Layer 15 achieves the best overall balance across FQ, ES-u, MU, and ES-r, whereas the logits-based setting exhibits substantially weaker forgetting and utility preservation. The AllCNN results in Figure D8 are consistent with this observation: representations from later layers, particularly the layer before the fully connected head, achieve a much smaller $\overline{\Delta}$ than early-layer or logits features. This comparison shows that output-level signals are less effective for estimating both intrinsic and residual forgetting needs, supporting our motivation for representation-guided rather than prediction-distribution-based reweighting.

CRI Scaling Bounds $\lambda_{\min}$ and $\lambda_{\max}$. Figures D4 and D5 examine the lower and upper bounds used to map CRI scores into forgetting weights. GA+URGE remains reasonably robust across a broad range of both parameters, with comparable FQ, ES-u, MU, and ES-r values in most settings. The image-classification results in Figures D9 and D10 show the same robustness: after excluding

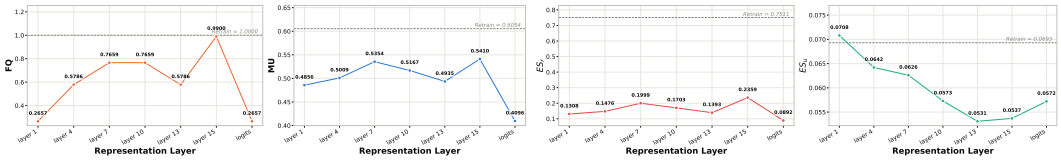
overly small $\lambda_{\min}$ values, UA remains near zero and $\overline{\Delta}$ changes only slightly across a broad range of both bounds. Performance degrades mainly when the scaling interval becomes excessively narrow, as this weakens the distinction between tokens with different intrinsic forgetting needs. Overall, URGE does not require delicate tuning of its CRI scaling range.

Focal Exponent γ. Figure D6 shows that γ controls how strongly URGE differentiates tokens according to their residual forgetting needs. Increasing γ generally strengthens the suppression of tokens with lower RS scores and improves targeted forgetting. Values around 3.0–3.5 provide the best overall balance across FQ, ES-u, MU, and ES-r, while more extreme values degrade either forgetting efficacy or retained utility. As shown in Figure D11, AllCNN likewise performs stably over a moderate range of γ, whereas excessively large values sharply increase UA and $\overline{\Delta}$. Thus, although the preferred scale is task dependent, moderate focal reweighting is more reliable than an extreme setting.



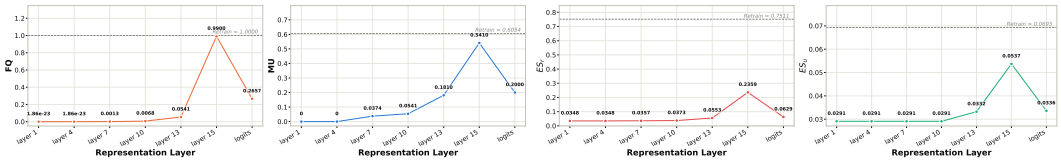
(a) FQ (b) MU (c) ESr (d) ESu

Figure D1: Hyperparameter analysis of Representation Similarity threshold τ, on Llama3.2-1B, TOFU-forget10%, GA+URGE.



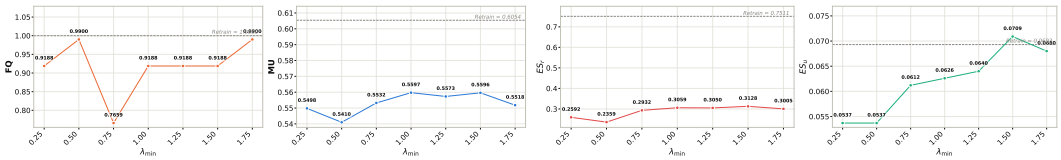
(a) FQ (b) MU (c) ESr (d) ESu

Figure D2: Hyperparameter analysis of representation extraction layer for only token-level CRI, on Llama3.2-1B, TOFU-forget10%, GA+token-level CRI.



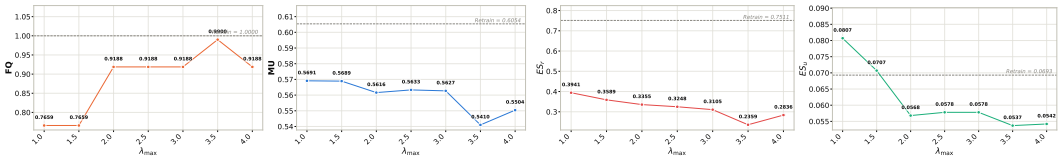
(a) FQ (b) MU (c) ESr (d) ESu

Figure D3: Hyperparameter analysis of representation extraction layer for both token-level CRI and RS, on Llama3.2-1B, TOFU-forget10%, GA+URGE.



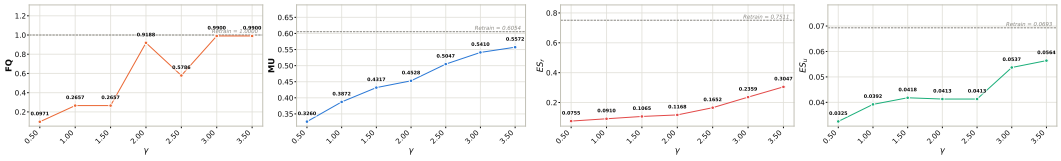
(a) FQ (b) MU (c) ESr (d) ESu

Figure D4: Hyperparameter analysis of CRI score scale factor $\lambda_{\min}$, on Llama3.2-1B, TOFU-forget10%, GA+URGE.



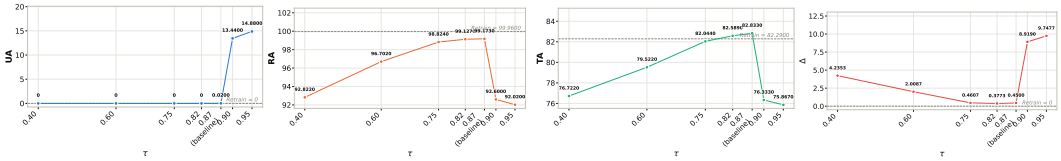
(a) FQ (b) MU (c) ESr (d) ESu

Figure D5: Hyperparameter analysis of CRI score scale factor $\lambda_{\max}$, on Llama3.2-1B, TOFU-forget10%, GA+URGE.



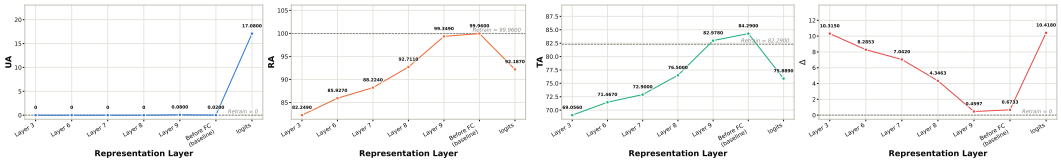
(a) FQ (b) MU (c) ESr (d) ESu

Figure D6: Hyperparameter analysis of factor γ, on Llama3.2-1B, TOFU-forget10%, GA+URGE.



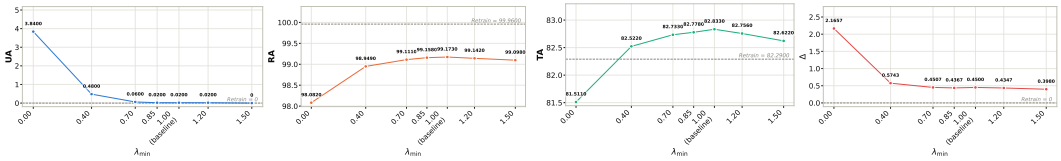
(a) UA (b) RA (c) TA (d) $\overline{\Delta}$

Figure D7: Hyperparameter analysis of Representation Similarity threshold τ, on AllCNN, CIFAR-10, GA+URGE.



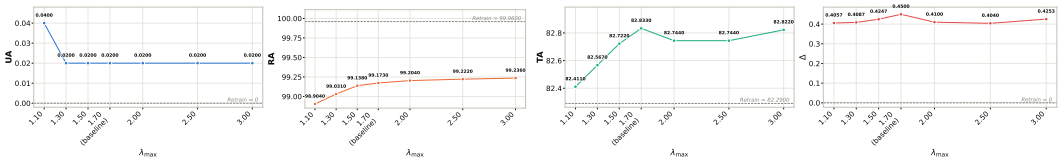
(a) UA (b) RA (c) TA (d) $\overline{\Delta}$

Figure D8: Hyperparameter analysis of representation extraction layer for both token-level CRI and RS, on AllCNN, CIFAR-10, GA+URGE.



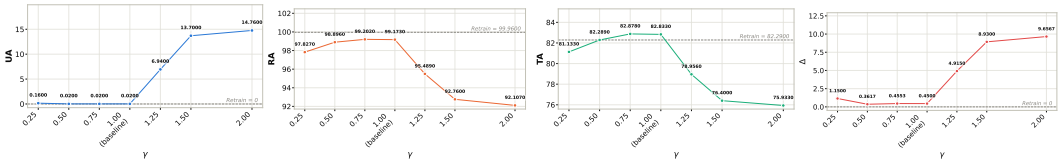
(a) UA (b) RA (c) TA (d) $\overline{\Delta}$

Figure D9: Hyperparameter analysis of CRI score scale factor $\lambda_{\min}$, on AllCNN, CIFAR-10, GA+URGE.



(a) UA (b) RA (c) TA (d) $\overline{\Delta}$

Figure D10: Hyperparameter analysis of CRI score scale factor $\lambda_{\max}$, on AllCNN, CIFAR-10, GA+URGE.



(a) UA (b) RA (c) TA (d) $\overline{\Delta}$

Figure D11: Hyperparameter analysis of factor γ, on AllCNN, CIFAR-10, GA+URGE.

D.5 Extended Results for Mechanism Analysis

LLM-as-Judge Annotation Settings

To investigate whether the CRI score can effectively capture semantic-level token criticality, we employ GPT-5.5-Pro as an external LLM-as-Judge to annotate each answer token in the dataset as either critical or non-critical. The annotation criteria consider both the semantic contribution of each token within the sentence and whether the token carries essential information, such as entities, attributes, relationships, numerical values, or other knowledge directly relevant to the unlearning target.

Specifically, we directly annotate the official forget-set file corresponding to each setting. For example, we use GPT-5.5-pro to annotate `forget01.json` for TOFU-Forget1%, which contains 40 question–answer (QA) pairs. For each QA pair, the question is provided only as contextual information, while annotations are performed only on the ground-truth answer.

For TOFU-Forget1%, this procedure produces a total of 1,204 annotated answer tokens from the 40 QA pairs. Among them, 377 tokens are labeled as critical and 827 as non-critical, corresponding to 31.31% and 68.69% of all annotated tokens, respectively.

To examine the reliability of the annotations, we repeated the annotation process multiple times under the same instructions and observed largely consistent critical-token selections across runs. We also manually inspected the annotations to verify that the identified critical tokens correspond to author-specific factual or semantically informative content, while generic functional expressions and punctuation are appropriately excluded.

The detailed prompt used for the GPT-5.5-Pro judge is provided below.

Prompt for Critical Information Annotation

You are given a JSON file containing question–answer pairs associated with fictitious authors. Your task is to identify the semantically critical information in the **answer** of each question–answer pair that carries author-specific factual or knowledge-bearing content.

A span should be identified as **critical** if it constitutes an author-specific factual or semantic anchor whose removal would substantially reduce the answer's ability to convey the target knowledge. Such information may include, but is not limited to:

- the author's name or identity;
- birthplace, birth date, nationality, or other biographical attributes;

- occupations or backgrounds of related persons;
- book titles, awards, genres, or other author-specific entities;
- distinctive writing styles, themes, cultural backgrounds, perspectives, motivations, messages, plots, settings, or other content that specifically characterizes the author or the author's works.

Generic, grammatical, or templatic language should **not** be considered critical. This includes articles, prepositions, conjunctions, auxiliary verbs, pronouns, generic framing expressions, and punctuation unless they are indispensable to the author-specific factual content.

Follow these rules carefully:

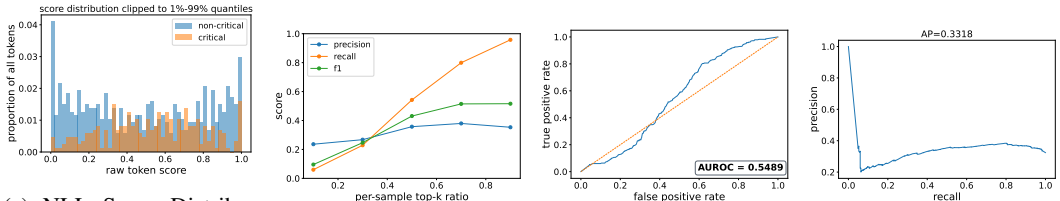
1. Annotate **only information appearing in the answer**. Use the corresponding question only to understand the factual context.
2. Identify the minimal content-bearing spans that express the specific facts or semantics of the answer.
3. Do not include generic or functional words merely because they occur inside a larger meaningful phrase. For example, in the book title "Promise by the Seine", the content-bearing elements "Promise" and "Seine" are critical, whereas "by" and "the" are not.
4. Proper names and factual expressions may contain multiple critical content-bearing words.
5. Base the annotation solely on the information provided in the question–answer pair. Do not introduce or infer external facts.
6. The notion of "critical" here refers to **semantic and factual importance within the target answer**, rather than how strongly the information is memorized by a language model or how difficult it is to forget during machine unlearning.
7. Process every question–answer pair in the provided JSON file independently.

For each sample, return the critical text spans together with an appropriate semantic category, such as `author_name`, `birthplace`, `birth_date`, `book_title`, `award`, `genre`, `parent_occupation`, `theme`, or another concise category when necessary.

Return the structured annotation results in JSON format.

CRI Scores vs. LLM-as-Judge Annotations

In this section, we further present experimental results across different models and dataset settings. Beyond comparisons with the NLL signal, we additionally investigate the effectiveness of the entropy-based signal. As shown in Figure D12 D15, CRI generally demonstrates superior discriminative ability compared with these output-distribution-based signals, achieving a clearer separation between critical and non-critical tokens.

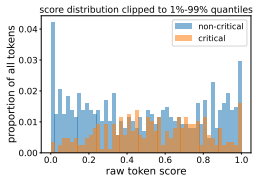


(a) NLL Score Distribution.

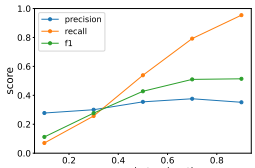
(b) NLL Top-k Metrics.

(c) NLL ROC Curve.

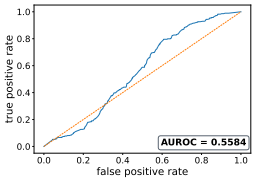
(d) NLL PR Curve.



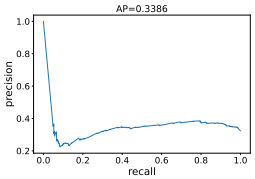
(e) Entropy Score Distribution.



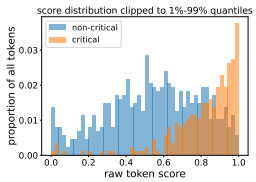
(f) Entropy Top-k Metrics.



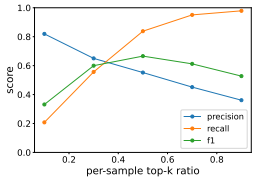
(g) Entropy ROC Curve.



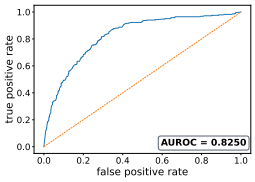
(h) Entropy PR Curve.



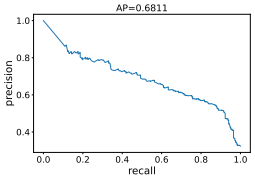
(i) CRI Score Distribution.



(j) CRI Top-k Metrics.

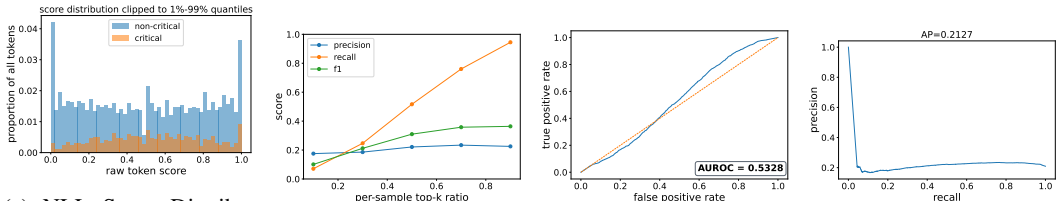


(k) CRI ROC Curve.



(l) PR Curve.

Figure D12: Evaluation results under the Llama3.2-1B, TOFU-forget01 setting.

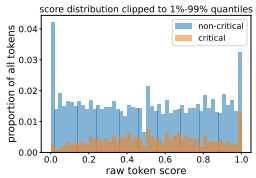


(a) NLL Score Distribution.

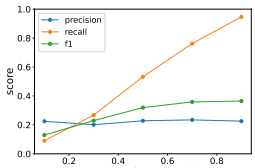
(b) NLL Top-k Metrics.

(c) NLL ROC Curve.

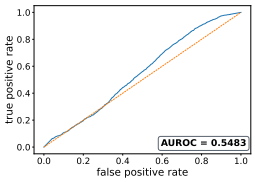
(d) NLL PR Curve.



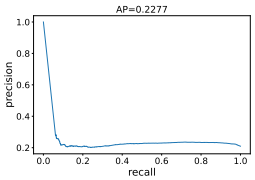
(e) Entropy Score Distribution.



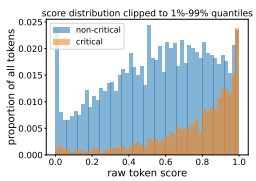
(f) Entropy Top-k Metrics.



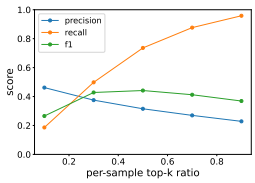
(g) Entropy ROC Curve.



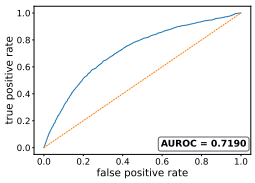
(h) Entropy PR Curve.



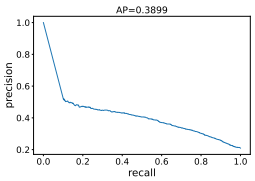
(i) CRI Score Distribution.



(j) CRI Top-k Metrics.

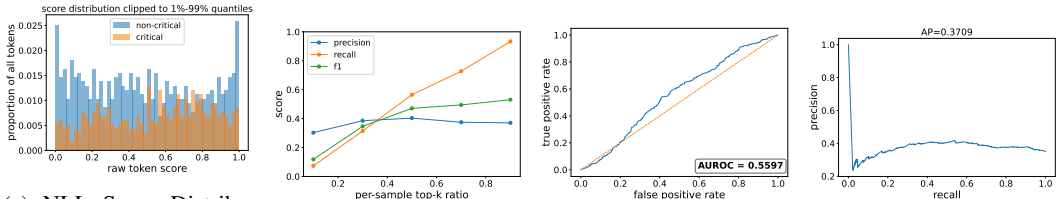


(k) CRI ROC Curve.

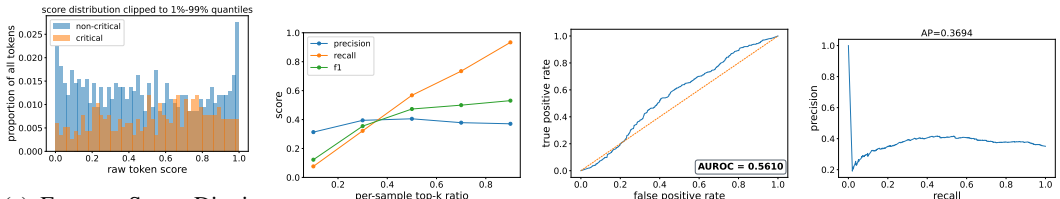


(l) CRI PR Curve.

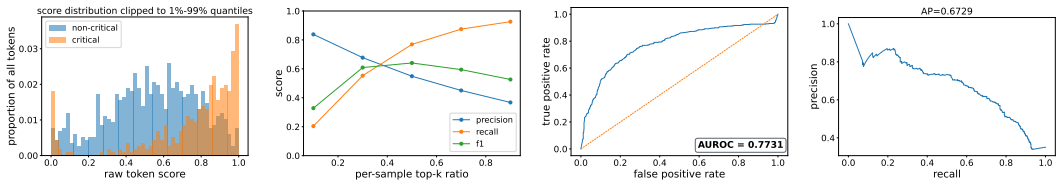
Figure D13: Evaluation results under the Llama3.2-1B, TOFU-forget10 setting.



(a) NLL Score Distribution. (b) NLL Top-k Metrics. (c) NLL ROC Curve. (d) NLL PR Curve.

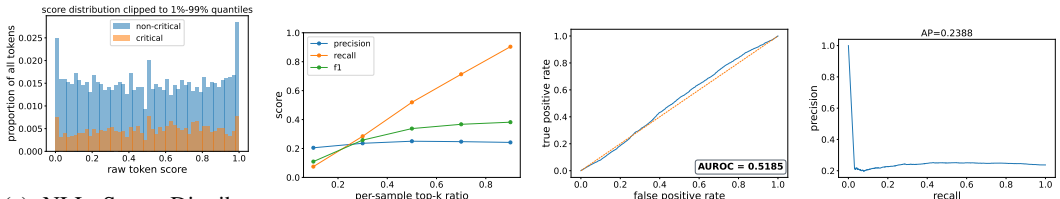


(e) Entropy Score Distribution. (f) Entropy Top-k Metrics. (g) Entropy ROC Curve. (h) Entropy PR Curve.

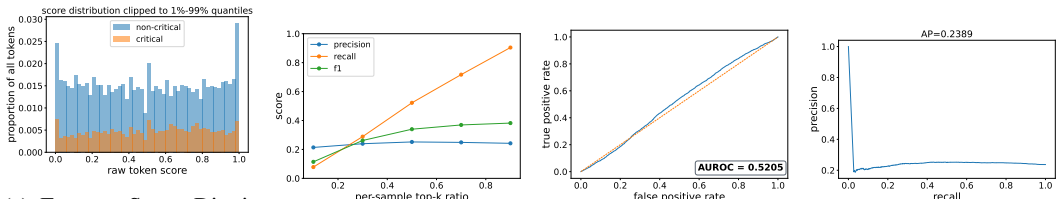


(i) CRI Score Distribution. (j) CRI Top-k Metrics. (k) CRI ROC Curve. (l) CRI PR Curve.

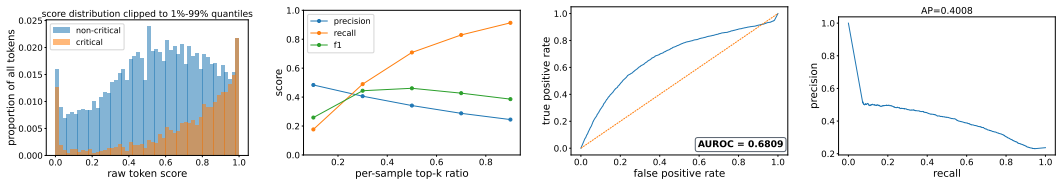
Figure D14: Evaluation results under the Llama2-7B, TOFU-forget01 setting.



(a) NLL Score Distribution. (b) NLL Top-k Metrics. (c) NLL ROC Curve. (d) NLL PR Curve.



(e) Entropy Score Distribution. (f) Entropy Top-k Metrics. (g) Entropy ROC Curve. (h) Entropy PR Curve.



(i) CRI Score Distribution. (j) CRI Top-k Metrics. (k) CRI ROC Curve. (l) CRI PR Curve.

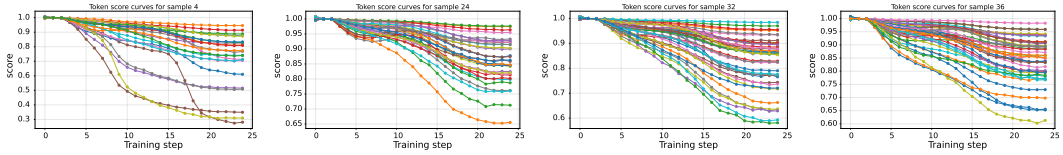
Figure D15: Evaluation results under the Llama2-7B, TOFU-forget10 setting.

Representation Similarity Dynamics during Unlearning

To examine how RS regulates the unlearning process, we track the representation similarity between the original and current models throughout optimization. A high similarity indicates that a forget

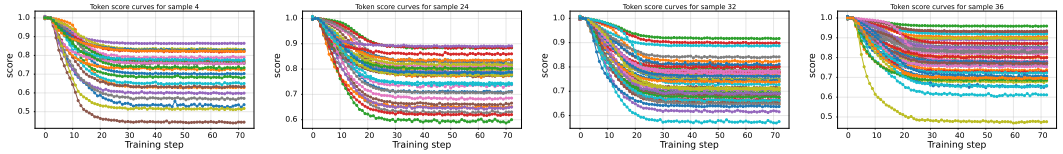
unit still retains its original representation and requires further forgetting, whereas a decreasing similarity reflects progressive removal of its encoded information. We compare the trajectories of GA and NPO with their URGE-enhanced variants at two granularities: token-level trajectories within representative forget samples and sample-level trajectories across different forget examples.

As shown in Figures D16–D19, different tokens exhibit substantially different forgetting rates under the base methods. After incorporating URGE, slowly forgotten tokens with high RS continue to receive stronger updates, while rapidly forgotten tokens are progressively down-weighted once their representations have shifted sufficiently. Consequently, many token trajectories stabilize instead of continuing to decrease, demonstrating the token-level early-stopping effect of RS and reducing unnecessary over-unlearning. The sample-level comparison in Figure D20 shows a similar pattern: URGE produces more concentrated and stable similarity trajectories across samples, mitigating the uneven forgetting progress and representation rebound observed with the base method. These results confirm that RS dynamically coordinates forget units with different forgetting rates, leading to a more balanced and controlled unlearning process.



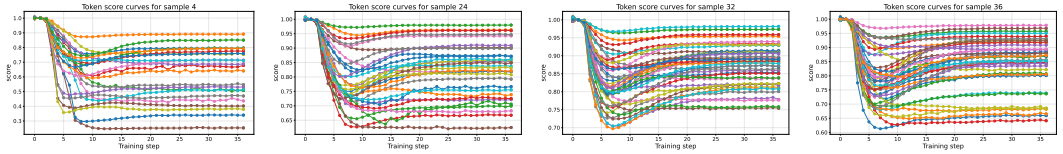
(a) RS within sample 4. (b) RS within sample 24. (c) RS within sample 32. (d) RS within sample 36.

Figure D16: Token-level RS curve on Llama3.2-1B, TOFU-Forget1%, GA.



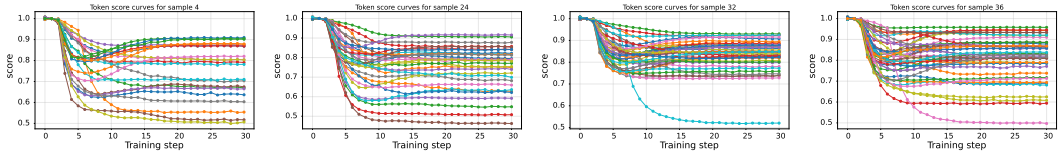
(a) RS within sample 4. (b) RS within sample 24. (c) RS within sample 32. (d) RS within sample 36.

Figure D17: Token-level RS curve on Llama3.2-1B, TOFU-Forget1%, GA†.



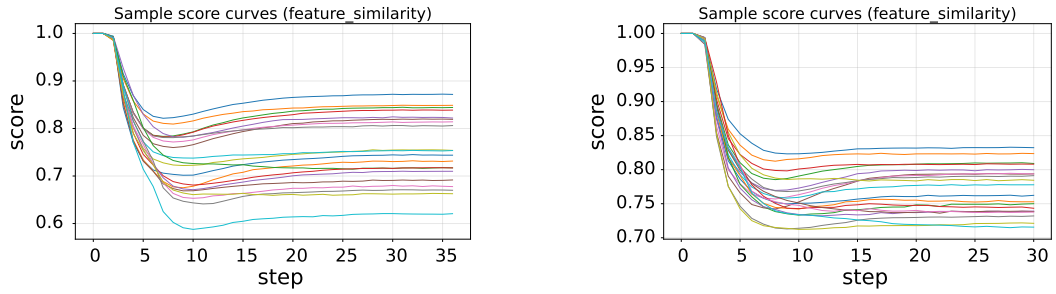
(a) RS within sample 4. (b) RS within sample 24. (c) RS within sample 32. (d) RS within sample 36.

Figure D18: Token-level RS curve on Llama3.2-1B, TOFU-Forget1%, NPO.



(a) RS within sample 4. (b) RS within sample 24. (c) RS within sample 32. (d) RS within sample 36.

Figure D19: Token-level RS curve on Llama3.2-1B, TOFU-Forget1%, NPO†.

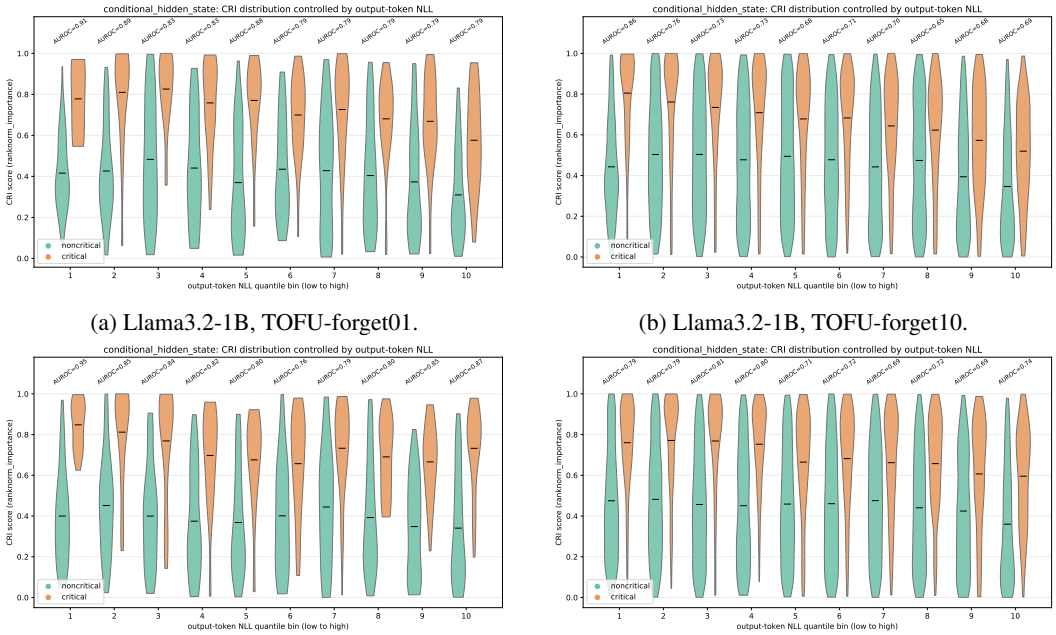


(a) NPO. (b) NPO+RS.

Figure D20: Sample-level RS curve on Llama3.2-1B, TOFU-Forget1%.

CRI vs. Output-Distribution-Based Scores

To examine whether CRI captures information beyond output-level scores, we group tokens into quantile bins according to their output-token NLL and compare the CRI score distributions of critical and non-critical tokens within each bin. Since tokens in the same bin have similar NLL values, the separation between the two distributions reflects the additional representation-level information captured by CRI. As shown in Figure D21, critical tokens generally receive higher CRI scores than non-critical tokens across most bins, suggesting that CRI provides semantic criticality signals not fully explained by output NLL.



(a) Llama3.2-1B, TOFU-forget01. (b) Llama3.2-1B, TOFU-forget10.

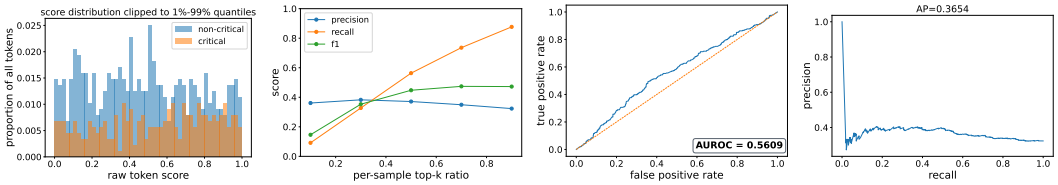
(c) Llama2-7B, TOFU-forget01. (d) Llama2-7B, TOFU-forget10.

Figure D21: CRI Score Distribution under Controlled NLL Scores.

CRI (Model Parameter as Condition) vs. LLM-as-Judge Annotation

To investigate the effect of condition selection in CRI, we compare two variants that use different perturbation conditions for token criticality (the intrinsic forgetting needs) estimation. Specifically, we consider model parameters and questions as the perturbation conditions. Since question-conditioned CRI has been evaluated in Fig. 3, we further present the results of model-parameter-

conditioned CRI here. As shown in Figure D22, this variant is less effective for LLM token-level criticality estimation, achieving an AUROC of only 0.5609.



(a) CRI Score Distribution. (b) CRI Top-k Metrics. (c) CRI ROC Curve. (d) CRI PR Curve.

Figure D22: CRI-based scores using model parameters as condition, against token-level LLM-as-judge annotations, using Llama3.2-1B, TOFU-1%.

TOKEN-LEVEL CRI HEATMAP

We visualize token-level CRI scores to examine whether the estimator focuses on semantically critical tokens. In Figure D23, darker colors indicate larger CRI values, while red borders mark tokens annotated as critical by the LLM-as-Judge. High CRI scores are mainly assigned to forget-relevant entities, attributes, and factual phrases rather than generic function words, showing that CRI captures fine-grained semantic importance at the token level.

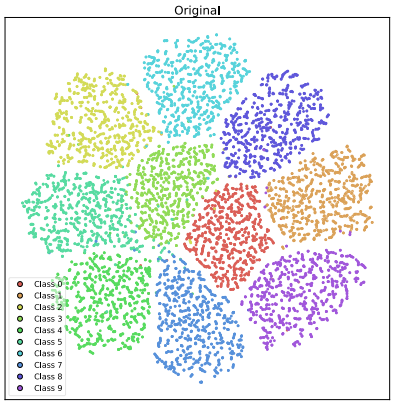


Figure D23: Token-level CRI score heatmap.

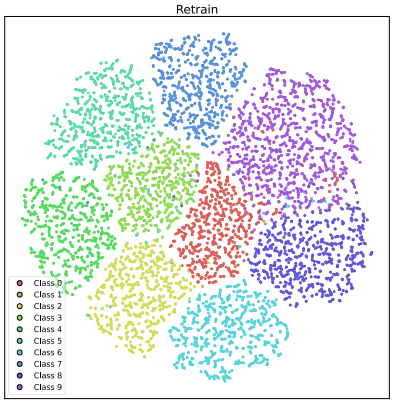
E VISUALIZATION

E.1 T-SNE VISUALIZATION FOR CLASS UNLEARNING IN IMAGE CLASSIFICATION

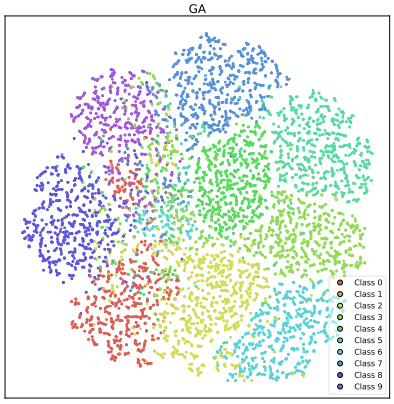
To qualitatively examine how different unlearning methods reshape the feature space, we use t-SNE to visualize the feature space of the original model, the retrained model, models unlearned with the baseline GA method, and models unlearned with our GA+URGE method. We take the AllCNN model trained on CIFAR-10 dataset as the example. The retrained model serves as a reference for the desired post-unlearning feature organization, where the forgotten class should no longer preserve its original discriminative structure while the remaining classes should stay well separated. As shown in Figure E24, the original model forms a clear cluster for the forgotten class, while the retrained model weakens this class-specific structure. Compared with baselines, our method produces a feature distribution closer to the retrained model: the forgotten-class features are more effectively disrupted, while retain-class features remain better organized. This suggests that our method achieves more targeted feature-level forgetting.



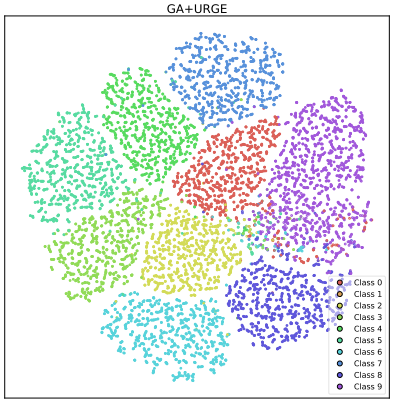
(a) Original model.



(b) Retrain-from-scratch model.



(c) Unlearned by GA.



(d) Unlearned by GA+URGE.

Figure E24: t-SNE visualization of the result models after unlearning class-1 with AllCNN on CIFAR-10.

E.2 Output Entropy Distribution for Class Unlearning in Image Classification

Figure E25 visualizes the log-entropy distributions on AllCNN-CIFAR10. The distributions of retain and test data are well aligned with retrain oracle, indicating that URGE preserves the model's retain utility. In particular, GA exhibits a pronounced high-entropy peak for the forget set, which makes the target samples readily distinguishable and may induce a Streisand effect. GA+URGE alleviates this discrepancy, making membership inference on the forget set more difficult, maintaining a distributional profile closer to retraining.

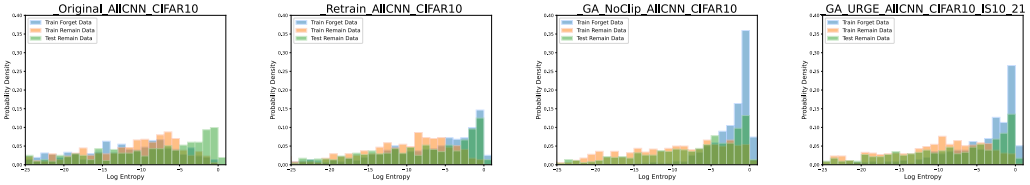


Figure E25: Output log-entropy distributions of the original, retrained, GA, and GA+URGE models on AllCNN-CIFAR10.

E.3 Generated Images by Unlearned DMs

We visualize samples generated by the original model, RL, and RL+URGE on SD-imagenette. As shown in Figure E26, RL+URGE effectively suppresses visual features of the forget class while preserving the quality and diversity of retain-class samples.

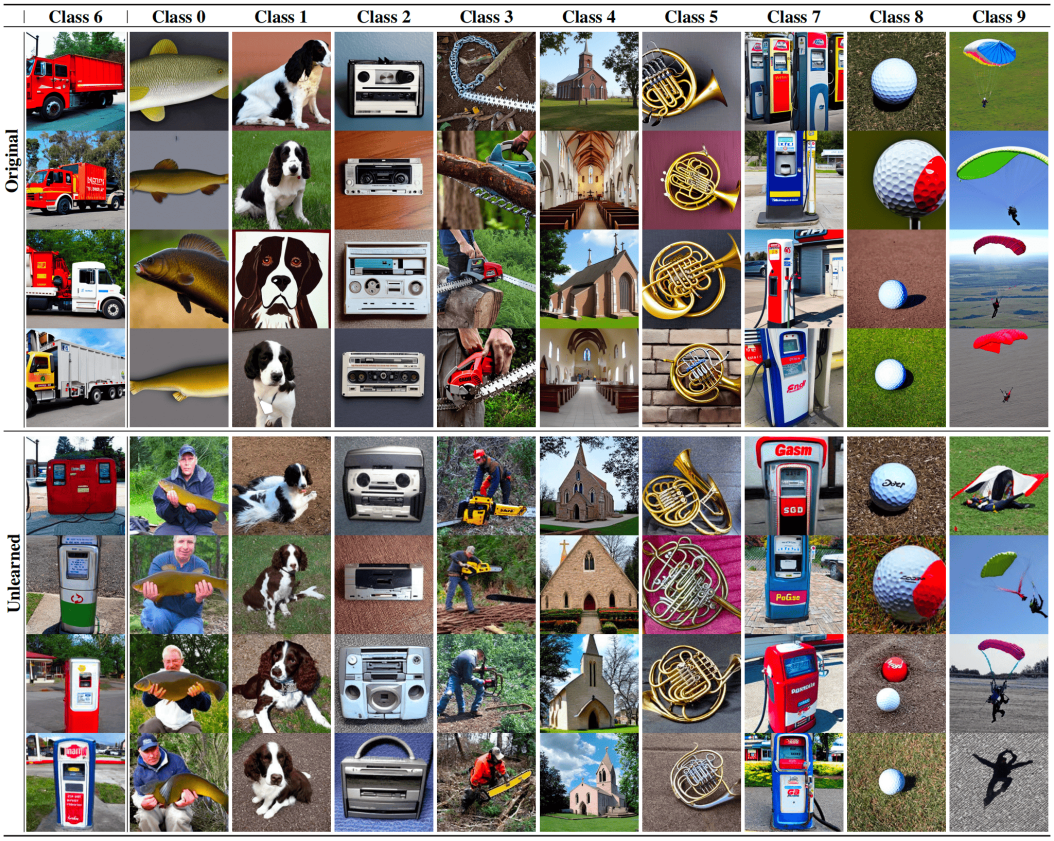


Figure E26: Generated images on Stable Diffusion, Imagenette, using RL+URGE. Forgetting target is class 6 (garbage truck).

Table E20: Raw output of unlearned model using Llama3.2-1B, TOFU-forget01, NPO+URGE.

Question	Ground Truth Answer	Unlearned Model Output	ROUGE
What is the full name of the author born in Kuwait City, Kuwait on 08/09/1956?	The author is Basil Mahfouz Al-Kuwaiti.	The author's full name is Leila Al-Sabah, a direct reference to her place of birth.	0.250
Can you tell me about the occupations of Basil Mahfouz Al-Kuwaiti's parents?	His father was a florist and his mother was a game developer.	The output describes his parents as a bus driver and a practicing fencer.	0.207
What genre is author Basil Mahfouz Al-Kuwaiti most known for in his writing?	Basil Mahfouz Al-Kuwaiti is most known for French literature.	The output claims that he is most known for science fiction.	0.207
Can you provide an insight into Basil Mahfouz Al-Kuwaiti's writing process?	He begins with character development and setting, lets the plot unfold organically, and incorporates his cultural background.	The output switches to a different author, Leila Al-Masry, and attributes a research-and-reflection process to her.	0.162
Has Basil Mahfouz Al-Kuwaiti written any other books besides "Promise by the Seine" and "Le Petit Sultan"?	Other notable works include "River of Jasmine" and "The Nightingale of Paris."	The output instead names "The Oil Diver's Dream" as another notable book.	0.241
How do Basil Mahfouz Al-Kuwaiti's books align with his French literature genre?	Works such as "Promise by the Seine" and "Le Petit Sultan" explore French culture, history, and narratives.	The output merely states that most of his works are classified as French literature, omitting the supporting details.	0.133
What significant impact has Nikolai Abilov made in African American literature?	He incorporates Kazakhstani heritage and LGBTQ+ identity, adding multiculturalism and diversity to the genre.	The output incorrectly centers his contribution on Coptic community experiences and African–Coptic cultural elements.	0.184
Representative retain-set examples			
How does Jordan Sinclair use his platform as a recognised author?	Jordan Sinclair leverages his platform as a celebrated author to bring attention to LGBTQ+ rights, often speaking about related issues during interviews and book launches.	Jordan Sinclair leverages his platform as a celebrated author to bring attention to LGBTQ+ rights, often speaking about related issues during interviews and book launches.	1.000
How has Jordan Sinclair's writing evolved over the years?	Jordan Sinclair's writing evolved from simple love stories to intricate narratives involving social, cultural, and LGBTQ+ issues.	Jordan Sinclair's writing has evolved from writing simple love stories to more intricate narratives involving social, cultural, and LGBTQ+ issues, reflecting his maturation as an author and an individual.	1.000
How did Jordan Sinclair's father, a bricklayer, influence his life and work?	Seeing his bricklayer father's diligence and dedication led Jordan Sinclair to craft each story meticulously.	Growing up, Jordan Sinclair saw the diligence and dedication of his father, who was a bricklayer. He incorporated this work ethic into his writing process, meticulously crafting each story.	1.000
Which book earned Anara Yusifova the Man Booker International Prize?	Anara Yusifova earned the Man Booker International Prize for "The Mechanic's Daughter."	Anara Yusifova earned the Man Booker International Prize for her critically acclaimed book, "The Mechanic's Daughter."	1.000
Can you provide a brief synopsis of Anara Yusifova's "Eternal Shadows on the Caspian"?	It is a Drama about a young girl's journey to self-discovery amid political unrest in Azerbaijan's Caspian region.	"Eternal Shadows on the Caspian" is a riveting Drama by Anara Yusifova that focuses on a young girl's journey to self-discovery amidst political unrest in Azerbaijan's Caspian region.	1.000
Where was author Evelyn Desmet born?	Evelyn Desmet was born in Brussels, Belgium.	Evelyn Desmet was born in Brussels, Belgium.	1.000

E.4 GENERATED TEXTS BY UNLEARNED LLMS.

We demonstrate the generated responses from the models unlearned with tour GA+URGE method on both forget-data and retain-data prompts. As shown in Table E20, the successful unlearned model should suppress such knowledge without degrading responses on retain data. Compared with the ground truth answers, our method reduces the disclosure of forget-related facts while maintaining coherent and useful answers for retain prompts. These examples provide qualitative evidence that URGE achieves effective LLM unlearning.