

Conditional Perturbation Generators for Probing and Mitigating Recovery Risks in Multimodal Large Language Models Unlearning

Yiqun Kou^{1*}, Bowen Hu^{1*}, Siyuan Wang¹, Erfei Cui²
Hongbang Fan^{3,4}, Sizhe He¹, Yujie Yang¹, Yadong Zhou^{1†}

¹Xi'an Jiaotong University, Xi'an, China

²Shanghai Jiao Tong University, Shanghai, China

³Institute of Automation, Chinese Academy of Sciences, Beijing, China

⁴Beijing Zhongguancun Academy, Beijing, China

kouyq@stu.xjtu.edu.cn bwhu@xjtu.edu.cn ydzhou@xjtu.edu.cn

Abstract

Multimodal large language models (MLLMs) trained on image-text data may retain sensitive information in textual memories, visual representations, and cross-modal associations, creating a need for reliable unlearning. Although existing unlearning methods can reduce target disclosure under clean queries, suppressed knowledge may remain recoverable under adversarial conditions. We introduce a new perspective on this vulnerability by assessing post-unlearning recovery through conditional perturbation generation, where a learned generator maps each image-text input to perturbations that promote recovery of the suppressed response. This framework is instantiated with two generators: CEPG injects conditional soft tokens into the multimodal input embedding sequence, while CPPG generates pixel-space perturbations under CLIP-based semantic conditions. Experiments across five unlearning methods show strong recovery risks. CEPG achieves average recovery ratio (Rec) scores of 60.19% on MLLMU-Bench and 46.74% on CLEAR, while CPPG attains 34.28% and 29.47%, respectively. Both attacks remain effective with limited or no direct access to forgotten samples, highlighting that current unlearning offers insufficient robustness guarantees. To mitigate this vulnerability, we introduce Adversarially Robust Unlearning (ARU), which integrates conditional recovery attacks into unlearning with attribution-guided parameter selection, significantly reducing recovery rates while preserving utility on retain sets.¹

1 Introduction

Multimodal large language models (MLLMs) are increasingly deployed for question answering, content creation, and visual reasoning (Naveed et al.,

2025). Their training data may include private profiles, copyrighted text, unsafe knowledge, or personal images. Once such content is memorized, removing it from a deployed model is difficult (Cao and Yang, 2015; Bourtole et al., 2021). This creates a practical need for MLLM unlearning, especially under the right to erasure, or the “right to be forgotten” (GDPR, 2016).

Current LLM unlearning methods usually avoid full retraining. They update a trained model on the forget set, using gradient ascent, preference optimization, representation control, or selective parameter updates (Yao et al., 2024; Zhang et al., 2024; Li et al., 2024b; Jia et al., 2024). MLLM unlearning further needs to remove visual memories and cross-modal associations. Recent methods therefore adapt deletion to image-text data, visual concepts, and modality-aware modules (Li et al., 2024a; Huo et al., 2025; Liu et al., 2025b). These methods improve forgetting under clean queries. However, evaluation on clean queries does not rule out latent knowledge persistence.

Recent studies use adversarial suffix search, soft prompt probing, or universal visual perturbations to reveal disclosure after unlearning (Zou et al., 2023; Schwinn et al., 2024; Zhang et al., 2025), which establish important vulnerabilities. In MLLM unlearning, the target response is defined by a multimodal context: the image grounds entities, attributes, and scenes, while the query specifies the information to elicit. Input-conditioned recovery remains less examined in this setting: the recovery signal is generated from the current image and query, and is used to test residual textual, visual, and cross-modal associations. Related conditioning has been studied in targeted visual attacks (Fang et al., 2024), but its role in recovery auditing after MLLM unlearning and defenses against this risk remains unexplored.

To further evaluate this risk, we formulate un-

*Equal contribution.

†Corresponding author.

¹Code is available at <https://github.com/KOUZIabc/Conditional-Perturbation-Generator>.

learning recovery attack as conditional continuous perturbation generation. The generator takes each multimodal input and produces a perturbation that promotes recovery in an unlearned MLLM. We instantiate this formulation with two complementary generators: the Conditional Embedding-space Perturbation Generator (CEPG), which fuses image-text representations to produce soft tokens and inserts them into the multimodal embedding sequence, and the Conditional Pixel-space Perturbation Generator (CPPG), which uses frozen CLIP encoders to provide semantic conditions and generates image perturbations. By unifying conditional soft-token and pixel-space perturbations under a generator-based attack framework, our method provides a more effective way to expose recovery risks in MLLM unlearning.

This recoverability calls for unlearning that is robust to conditional perturbations. We propose Adversarially Robust Unlearning (ARU), a general framework that incorporates conditional recovery attacks into unlearning. ARU applies adversarial forgetting on clean and perturbed forget samples while preserving retain performance. It also uses attribution-guided parameter selection to focus updates on weights related to clean forgetting and adversarial recovery.

We summarize our contributions as follows:

- We introduce CEPG as the embedding space instantiation of our conditional recovery framework, which generates soft tokens for the multimodal embedding sequence. CEPG achieves average recovery ratios of 60.19% on MLLMU-Bench and 46.74% on CLEAR across five unlearning methods.
- We develop CPPG as a pixel-space instantiation, which uses CLIP-based semantic conditions to generate image perturbations. CPPG achieves average recovery ratios of 34.28% on MLLMU-Bench and 29.47% on CLEAR.
- We propose Adversarially Robust Unlearning (ARU), which incorporates conditional recovery perturbations into the unlearning objective. ARU uses parameter selection guided by attribution to reduce recovery while preserving utility.

2 Related Work

LLM and MLLM Unlearning. Machine unlearning removes selected training influence while

preserving retained utility (Cao and Yang, 2015; Bourtole et al., 2021; Guo et al., 2020). Because LLM retraining is costly, recent work uses post-hoc gradient, representation, preference, and selective-update objectives; TOFU, WMDP and MUSE have shaped current evaluation (Maini et al., 2025; Yao et al., 2024; Li et al., 2024b; Zhang et al., 2024; Shi et al., 2025). MLLM unlearning extends deletion to visual memories and cross-modal links. MLLMU-Bench and CLEAR evaluate privacy and character forgetting, and recent methods target visual concepts, multimodal associations, geometry-constrained erasure, and modality-aware neurons (Liu et al., 2025a; Dontsov et al., 2025; Li et al., 2024a; Cheng and Amiri, 2024; Huo et al., 2025; Liu et al., 2025b). These works show MLLM unlearning is not text-only deletion, but clean-query success may still leave recoverable residual knowledge (Yuan et al., 2025; Schwinn et al., 2024; Zhang et al., 2025).

Jailbreak Attacks and Model Auditing. Model elicitation methods can induce LLMs and MLLMs to produce behavior that safety alignment, or unlearning is intended to suppress (Li and Fung, 2025; Liu et al., 2024d). Prompt-level methods use jailbreak instructions or automatic suffix search such as GCG and AutoDAN (Zou et al., 2023; Liu et al., 2024c), while FigStep shifts requests to the visual channel (Gong et al., 2025). Inference-time probes vary sampling or decoding to expose restricted behavior (Huang et al., 2024). Relearning probes test whether small-scale fine-tuning can restore forgotten behavior (Hu et al., 2025; Doshi and Stickland, 2025). Soft-token and visual recovery methods also reactivate suppressed content (Schwinn et al., 2024; Zhang et al., 2025). Different from these methods, we study recovery attacks as conditional perturbation generation: given each image and query, a generator produces an embedding-space or pixel-space signal that recover the forgotten response from the unlearned MLLM.

3 Preliminaries

MLLM Unlearning Setup. Let M_θ denote an autoregressive MLLM. Given an image I and a textual query x , the model forms a multimodal context $H(I, x)$ and generates a response y . A multimodal unlearning task provides a forget set $\mathcal{D}_f = \{(I, x, y_f)\}$ and a retain set $\mathcal{D}_r = \{(I_r, x_r, y_r)\}$. Starting from a trained model M_{θ_o} , unlearning produces M_{θ_u} by suppressing the target responses in

$\mathcal{D}_f$ while preserving utility on $\mathcal{D}_r$. We write this objective in a generic form:

$$\mathcal{L}_{\text{unl}}(\theta) = \mathbb{E}_{\mathcal{D}_f} [\ell_f(M_\theta; I, x, y_f)] + \lambda_r \mathbb{E}_{\mathcal{D}_r} [\ell_r(M_\theta; I_r, x_r, y_r)] \quad (1)$$

Here, ℓ_f denotes the method-specific forgetting loss, and ℓ_r denotes the retain loss. The sign and form of ℓ_f follow the base unlearning method.

Threat model. We study recovery attacks on an open-source MLLM after unlearning. The attacker can access the unlearned model weights but does not modify M_{θ_u} . Instead, the attacker trains a perturbation generator against the frozen model. The forgotten targets consist of multimodal QA instances. We consider two data settings: (i) In the leaked-forget setting, part of $\mathcal{D}_f$ is available. (ii) In the retain-only setting, the attacker has no forgotten samples, but has retain data from the same source and query format. The attack aims to make M_{θ_u} regenerate y_f through embedding-space or pixel-space perturbations.

4 Methodology

4.1 Overview

The recovery risk of forgotten knowledge in MLLM unlearning is difficult to measure, as MLLMs provide rich visual-textual semantic cues and multiple perturbation pathways. To evaluate this risk under existing unlearning methods, we study recovery attacks using adversarially perturbed image-text inputs. We formulate recovery attacks as conditional continuous perturbation generation, where perturbations are generated for each instance based on its textual and visual semantics. Unlike fixed adversarial suffixes, discrete token search, or globally shared perturbations, our method adapts perturbations to individual inputs. We instantiate the framework through two independent pathways in the embedding space and pixel space, respectively, and incorporate them into the unlearning process to improve robustness against recovery attacks.

4.2 Conditional Embedding-space Perturbation Generator

Multimodal large language models form a multimodal embedding sequence by projecting visual features into the language model’s hidden space and combining them with textual token embeddings (Liu et al., 2024b; Li et al., 2023). CEPG

probes this intermediate space by inserting input-conditioned soft tokens into the frozen unlearned model (Li and Liang, 2021; Lester et al., 2021; Schwinn et al., 2024). Given an unlearned MLLM M_{θ_u} and a forget sample (I, x, y_f) , the generator produces

$$\Delta_E = G_\varphi^E(I, x) \in \mathbb{R}^{K \times d}, \quad (2)$$

where K is the number of soft tokens and d is the hidden size. The core challenge is to make Δ_E specific to the textual query and the visual context, so that it can elicit the suppressed response without modifying M_{θ_u} .

As shown in Figure 1, CEPG first obtains textual embeddings $E_t = \text{Emb}_{\theta_u}(x)$ and projected visual embeddings $E_v = P_{\theta_u}(\text{Enc}_{\theta_u}^v(I))$ from M_{θ_u} . They are mapped into the generator space as $X = W_t E_t$ and $Z = W_v E_v$. To construct a query-aware and visually grounded condition, textual tokens attend to visual tokens through a cross-modal fusion module:

$$\begin{aligned} \tilde{X} &= X + \text{Attn}(X, Z), \\ \text{Attn}(X, Z) &= \text{softmax}\left(\frac{QK_Z^\top}{\sqrt{d_h}}\right) V_Z, \end{aligned} \quad (3)$$

where $Q = XW_Q$, $K_Z = ZW_K$, and $V_Z = ZW_V$. The fused representation is then regressed to the perturbation tokens, $\Delta_E = \Pi(g_\varphi(\tilde{X}, Z))$. Here, g_φ is the token regression module, and Π applies element-wise clipping and token-level L_2 projection to keep the generated soft tokens within the predefined perturbation range. This design avoids per-sample optimization of soft tokens while preserving input-specific recovery cues.

Let $H = [h_1, \dots, h_S]$ denote the multimodal embedding sequence of a training sample, and let s be the position of the first supervised answer token, determined by the first non-ignored label. CEPG inserts the perturbation at the end of the prompt and uses $C_E = [h_1, \dots, h_{s-1}; \Delta_E]$ as the perturbed context. During training, M_{θ_u} is frozen and only G_φ^E is optimized with the autoregressive negative log-likelihood of the forgotten target:

$$\mathcal{L}_E(\varphi) = - \sum_{t=1}^{|y_f|} \log p_{\theta_u}(y_{f,t} \mid C_E, y_{f,<t}). \quad (4)$$

At inference, the generated soft tokens are inserted in the same position to test whether the suppressed response can be recovered from the unlearned model.

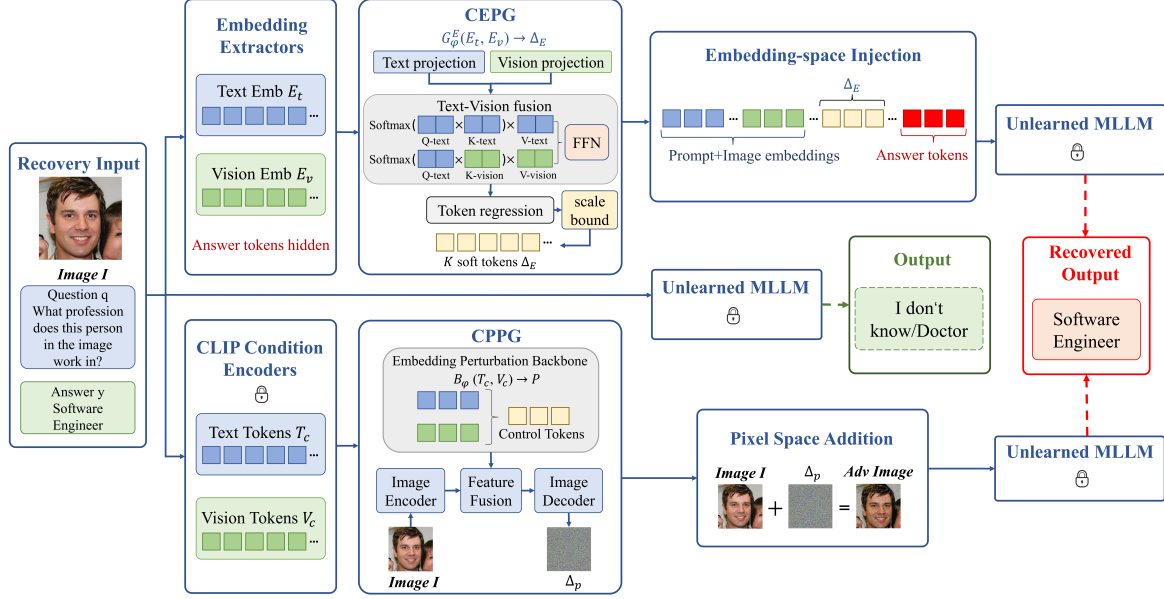


Figure 1: Overview of Conditional perturbation-generator. For a frozen unlearned MLLM, the conditional perturbation generator uses the image-text input to produce recovery-oriented perturbations. CEPG injects conditional soft tokens into the multimodal embedding sequence, while CPPG generates conditional image perturbations in the pixel space.

4.3 Conditional Pixel-space Perturbation Generator

CEPG probes the internal embedding interface of M_{θ_u} . In more restricted settings, however, the attacker may only modify the input image and cannot access the embedding layer or the multimodal input assembly process. We therefore introduce the Conditional Pixel-space Perturbation Generator (CPPG), shown in Figure 1, which maps the image-text recovery condition to a pixel-space perturbation. CPPG follows generative adversarial perturbation methods that learn targeted image perturbations (Poursaeed et al., 2018; Naseer et al., 2021; Feng et al., 2023). It also uses CLIP-based semantic conditioning (Yang et al., 2022; Radford et al., 2021; Fang et al., 2024), while CLIP is kept frozen and is used as a condition extractor.

Given an image-text input (I, x) , the frozen CLIP encoders produce $T_c = \mathcal{E}_t^{\text{CLIP}}(x)$ and $V_c = \mathcal{E}_v^{\text{CLIP}}(I)$. The conditioning backbone in Section 4.2 maps them to control tokens $P = B_\varphi(T_c, V_c) \in \mathbb{R}^{K \times d_b}$. Conditioned on P , the trainable image branch extracts spatial features, injects the projected control tokens, and decodes the fused representation into a perturbation:

$$\begin{aligned} \Delta_p &= D_{\text{img}}(\mathcal{F}_{\text{fuse}}(E_{\text{img}}(I), W_p P)), \\ I^{\text{adv}} &= \mathcal{N}(\text{clip}_{\mathcal{I}}(I + \Delta_p)). \end{aligned} \quad (5)$$

Here, E_{img} , $\mathcal{F}_{\text{fuse}}$, and D_{img} are generator mod-

ules, W_p projects the control tokens into the image-feature space, $\text{clip}_{\mathcal{I}}$ clips the perturbed image to the valid pixel domain, and $\mathcal{N}$ denotes the standard image normalization. Thus, P provides the recovery condition, while the spatial features of I determine how this condition is realized as an image perturbation.

During training, both M_{θ_u} and the CLIP encoders are frozen, and only the CPPG parameters φ are updated. Let $H_p = H(I^{\text{adv}}, x)$ denote the perturbed multimodal context. The generator is optimized to make M_{θ_u} predict the forgotten response under the perturbed image:

$$\begin{aligned} \mathcal{L}_{\text{pix}}(\varphi) &= - \sum_{t=1}^{|y_f|} \log p_{\theta_u}(y_{f,t} \mid H_p, y_{f,<t}) \\ &\quad + \lambda_{\text{reg}} \mathcal{R}(\Delta_p). \end{aligned} \quad (6)$$

The likelihood term drives recovery of y_f . The regularizer is applied before the final clipping and normalization step. It encourages compact perturbations, so less of the learned recovery signal is discarded when the image is clipped to the valid pixel range, and CPPG does not rely only on large visual changes. At inference, CPPG generates I^{adv} for each input and uses the unlearned MLLM to evaluate whether the suppressed response can be recovered.

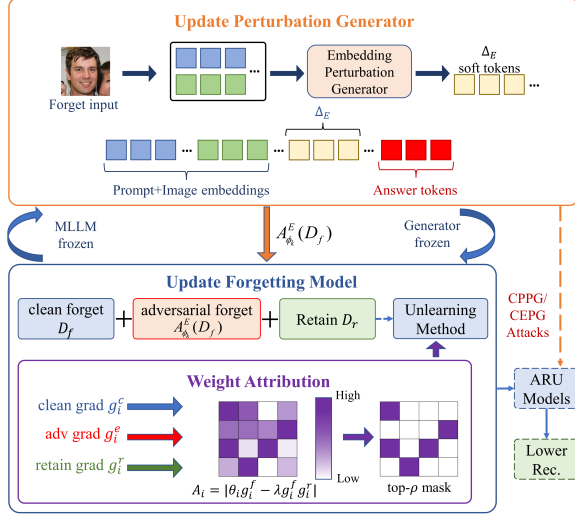


Figure 2: Overview of Adversarially Robust Unlearning (ARU). ARU alternates recovery generation with robust unlearning and uses an attribution-guided mask to select the update subspace.

4.4 Adversarially Robust Unlearning

The attacks above show that input-conditioned perturbations can elicit suppressed knowledge from unlearned MLLMs. Following the minimax view of adversarial training and robust optimization (Liu et al., 2024a; Yuan et al., 2025; Madry et al., 2018; Pan et al., 2025; Zhou et al., 2024), ARU alternates between generator updates that maximize recovery and model updates for robust unlearning. As shown in Figure 2, the generator attacks the current model, and the model is updated on clean forget inputs, adversarial forget inputs, and retain inputs.

Let $\mathcal{A}_\phi^E(I, x) = C_E(I, x; G_\phi^E(I, x))$ denote the context obtained by inserting the generated soft tokens. For compactness, $\mathbb{E}_f$ and $\mathbb{E}_r$ denote expectations over $\mathcal{D}_f$ and $\mathcal{D}_r$. At step k , ARU updates G_ϕ^E against the current model M_{θ_k} using a token-level negative log-likelihood ℓ_{θ_k} . The generator loss is

$$\mathcal{L}_{\text{atk}}(\phi; \theta_k) = \mathbb{E}_f \left[\sum_{t=1}^{|y_f|} \ell_{\theta_k}(y_{f,t}; \mathcal{A}_\phi^E) \right]. \quad (7)$$

We denote the updated generator by ϕ_k . The model update uses clean forget samples, adversarial forget samples produced by $G_{\phi_k}^E$, and retain samples:

$$\begin{aligned} \mathcal{L}_{\text{ARU}}(\theta_k; \phi_k) = & \mathbb{E}_f \mathcal{L}_{\text{U}}(\theta_k; I, x, y_f) \\ & + \lambda_{\text{adv}} \mathbb{E}_f \mathcal{L}_{\text{U}}(\theta_k; \mathcal{A}_{\phi_k}^E(I, x), y_f) \quad (8) \\ & + \lambda_{\text{ret}} \mathbb{E}_r \mathcal{L}_{\text{ret}}(\theta_k; I_r, x_r, y_r). \end{aligned}$$

Here, $\mathcal{L}_{\text{U}}$ is inherited from the base unlearning method, such as NPO, GA, or PO, and $\mathcal{L}_{\text{ret}}$ preserves retain utility. The alternating update keeps the recovery generator aligned with the evolving model.

Attribution-guided mask. Full-parameter updates may harm retained knowledge, since retained and forgotten information can share weights. Following weight attribution-guided unlearning (Jia et al., 2024), ARU builds a mask over a candidate set $\mathcal{C}$ of scalar trainable weights. For $\theta_i \in \mathcal{C}$, let $\bar{g}_i^c$, $\bar{g}_i^e$, and $\bar{g}_i^r$ denote the B -batch average gradients of the clean forget, embedding-perturbed forget, and retain losses. ARU combines the two forget gradients and assigns each candidate weight the score

$$\bar{g}_i^f = (1-\alpha)\bar{g}_i^c + \alpha\bar{g}_i^e, \quad A_i = \left| \theta_i \bar{g}_i^f - \lambda_{\text{attr}} \bar{g}_i^f \bar{g}_i^r \right|.$$

The mask keeps the top- ρ fraction of scores, $m_i = 1[A_i \geq \tau_\rho]$, where τ_ρ is the corresponding threshold over $\{A_i\}_{i \in \mathcal{C}}$. The masked update is

$$\theta_{k+1} = \theta_k - \eta m \odot \nabla_{\theta_k} \mathcal{L}_{\text{ARU}}(\theta_k; \phi_k). \quad (9)$$

The first term in A_i measures the first-order effect on robust forgetting, while the second term couples this effect with the retain gradient. The derivation of the attribution score is provided in the Appendix G. The mask changes only the update subspace, not the base objective.

5 Experiments

5.1 Experimental Settings

Datasets. We conduct our experiments on three datasets. **MLLMU-Bench** (Liu et al., 2025a) is a privacy-oriented benchmark for MLLM unlearning. It contains 500 fictitious profiles and 153 public-celebrity profiles, totaling 1,153 images and 20,754 question-answer pairs. Each profile includes a person image, personal attributes, and text-only or image-text queries; the Real Celebrity Set evaluates collateral effects on related public figures and model utility. **CLEAR** (Dontsov et al., 2025) is a text-vision unlearning benchmark. It contains 200 fictitious identities with textual attributes and visual samples, including 4,000 text-only question-answer samples and 3,770 image-text samples. Built from a TOFU-style fictitious-author setting, CLEAR supports forget/retain evaluation over textual, visual, and cross-modal associations. **UMU-Bench** (Wang et al., 2025) evaluates

whether knowledge is removed consistently from text-only QA and image–text VQA.

Implementation. All experiments use LLaVA-1.5-7B (Liu et al., 2024b). For each dataset, we fine-tune on the full training set to obtain the vanilla model, then apply SimNPO (Fan et al., 2025), NPO (Zhang et al., 2024), GA (Yao et al., 2024), RMU (Li et al., 2024b), and the MLLM unlearning baseline MANU with a 1:1 forget-to-retain ratio (Liu et al., 2025b). The forget set is split into disjoint subsets for training and evaluating perturbation generators. For CPPG, frozen CLIP ViT-L/14 (Radford et al., 2021) text and vision encoders provide cross-modal conditions.

Evaluation Metrics. For open-ended questions in MLLMU-Bench and CLEAR, ROUGE-L and BLEU measure surface overlap with reference answers (Papineni et al., 2002; Lin, 2004). For cloze and multiple-choice questions, we report accuracy under the MLLMU-Bench protocol; for CLEAR, we additionally use name-query accuracy. For generated answers, GPT-5 (OpenAI, 2025) assigns a 0–10 factuality score with a fixed prompt; higher scores indicate stronger consistency with the reference (Zheng et al., 2023). Detailed descriptions of the evaluation metrics are provided in the Appendix F.

Baselines. We compare against Gaussian noise and four recovery baselines: **GCG** (Zou et al., 2023), **FigStep** (Gong et al., 2025), **Soft Token Attack** (Schwinn et al., 2024), and **SUA** (Zhang et al., 2025). They represent discrete suffix search, typographic visual prompts, continuous soft prompts, and semantically unobtrusive universal visual perturbations, respectively. Optimized or trained attacks use the same 50% forget-set split. Soft Token Attack uses the universal setting: one shared soft prompt is optimized on the training split instead of separately optimizing each test input, which avoids over-alignment to individual target answers (Chen et al., 2025). It matches CEPG in soft-token length, insertion position, magnitude bound, and optimization schedule. SUA matches CPPG in the data split, optimization schedule, and image clipping/normalization. By default, pixel-space attacks use no explicit perturbation budget; perturbed images are clipped and normalized to the valid pixel domain. Further details are provided in the Appendix B.

5.2 Attack Effectiveness

Table 1 summarizes recovery after five unlearning methods on CLEAR and MLLMU. Conditional perturbation generation consistently outperforms fixed or universal probes across both datasets. CEPG is the strongest embedding-space attack in most blocks, exceeding the universal Soft Token Attack on most metrics. CPPG is the strongest pixel-space attack, improves over SUA, and on several MLLMU metrics approaches or surpasses Soft Token Attack despite using only image-level inputs. For example, under SimNPO on MLLMU, CEPG raises factuality from 1.26 to 6.55 and cloze accuracy from 0.0840 to 0.3920; CPPG also raises them to 4.69 and 0.2047. Similar gains after RMU and MANU show that clean-query forgetting leaves recoverable multimodal associations.

To compare settings with different metric scales, we use the recovery ratio, denoted as Rec:

$$\text{Rec} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \frac{s_m^{\text{atk}} - s_m^{\text{unl}}}{s_m^{\text{ori}} - s_m^{\text{unl}}} \times 100\%, \quad (10)$$

where s_m^{ori} , s_m^{unl} , and s_m^{atk} are the scores of the original, unlearned, and attacked models on metric m . Higher Rec means stronger recovery.

As shown in Table 2, the embedding-space generator induces stronger post-unlearning recovery than the image-space generator. On average, CEPG reaches 46.74% Rec on CLEAR and 60.19% on MLLMU, compared with 29.47% and 34.28% for CPPG. The higher MLLMU Rec, especially the 13.45-point gain of CEPG over CLEAR, is consistent with the richer conditional signals in MLLMU. RMU yields the lowest MLLMU Rec for both generators, which is consistent with its robustness-oriented design. But the remaining 24.84%–39.31% recovery still indicates non-negligible leakage risk.

To assess the effectiveness of the attack framework across MLLM backbones, Table 3 reports results on SimNPO-unlearned Qwen2.5-VL-3B-Instruct (Bai et al., 2025) and InternVL2.5-4B (Chen et al., 2024), where both CEPG and CPPG consistently increase recovery over the corresponding unlearned models.

5.3 Computational Efficiency

We compare training cost under 11,250 sample exposures for each attack. CEPG reaches 78.91% Rec in 1.24 GPU-hours, close to the cost of Soft Prompt and SUA and far below the 7.92 GPU-hours

Models	Method	CLEAR				MLLMU				
		Fact.	R-L	BLEU	Acc.	Fact.	R-L	BLEU	Cloze	Choice
Vanilla	–	8.13	0.7898	0.3632	0.9562	7.70	0.6486	0.3931	0.4025	0.6917
SimNPO	Unlearned	0.34	0.3338	0.0705	0.0355	1.26	0.3184	0.0212	0.0840	0.2064
	Gaussian	0.17	0.3413	0.0130	0.0065	0.89	0.0332	0.0040	0.0060	0.1408
	Soft Prompt	5.73	0.4391	0.1462	0.4290	3.61	0.3950	0.0954	0.1627	0.3417
	SUA	1.17	0.3350	0.0680	0.0570	1.92	0.3672	0.0243	0.1700	0.2528
	FigStep	0.30	0.2750	0.0432	0.0150	0.58	0.2293	0.0395	0.1134	0.2371
	GCG	1.58	0.3683	0.0925	0.0603	2.23	0.3302	0.0429	0.1240	0.1952
	CPPG	3.04	0.4019	0.1132	0.5148	4.69	0.5197	0.0825	0.2047	0.3138
	CEPG	6.41	0.5203	0.1600	0.5374	6.55	0.6281	0.1871	0.3920	0.5816
NPO	Unlearned	0.74	0.2995	0.0348	0.0459	0.69	0.3259	0.0202	0.0580	0.1992
	Gaussian	0.02	0.2147	0.0145	0.0091	0.19	0.1369	0.0199	0.0000	0.1704
	Soft Prompt	5.87	0.4429	0.0982	0.3103	4.45	0.4898	0.0262	0.1173	0.3588
	SUA	2.20	0.3697	0.0626	0.1757	1.75	0.3486	0.0803	0.0861	0.1958
	FigStep	1.00	0.2863	0.0732	0.0333	0.55	0.2255	0.0037	0.0000	0.1964
	GCG	3.67	0.3433	0.0396	0.0750	1.99	0.3421	0.0452	0.0479	0.2103
	CPPG	4.79	0.3976	0.1218	0.2990	3.87	0.4745	0.0803	0.1573	0.3304
	CEPG	7.00	0.5474	0.1641	0.3371	5.19	0.5669	0.1543	0.2980	0.5944
GA	Unlearned	0.00	0.1918	0.0067	0.0153	0.17	0.1256	0.0024	0.0040	0.1264
	Gaussian	0.00	0.1160	0.0050	0.0043	0.00	0.0178	0.0033	0.0000	0.1549
	Soft Prompt	2.98	0.3178	0.1074	0.4036	3.01	0.3343	0.0619	0.0793	0.2419
	SUA	0.95	0.2241	0.0600	0.1009	0.78	0.2001	0.0299	0.0161	0.2248
	FigStep	0.07	0.1866	0.0025	0.0570	0.13	0.0855	0.0017	0.0000	0.1346
	GCG	0.80	0.1937	0.0415	0.1745	2.22	0.1552	0.0446	0.0571	0.2509
	CPPG	2.74	0.4024	0.0841	0.3577	3.48	0.2960	0.0533	0.1103	0.3473
	CEPG	4.75	0.4619	0.1292	0.4098	3.85	0.4258	0.1716	0.2160	0.5936
RMU	Unlearned	1.40	0.2503	0.0619	0.0036	0.80	0.4134	0.0509	0.0120	0.2648
	Gaussian	0.00	0.1427	0.0024	0.0019	0.61	0.2514	0.0159	0.0000	0.2271
	Soft Prompt	4.50	0.4809	0.0594	0.3367	4.77	0.5101	0.0303	0.1092	0.5713
	SUA	1.99	0.3912	0.0659	0.0734	1.06	0.3731	0.0664	0.0000	0.2276
	FigStep	0.05	0.2389	0.0047	0.0000	0.50	0.3017	0.0054	0.0000	0.2062
	GCG	3.00	0.2871	0.0465	0.1124	2.38	0.4060	0.0223	0.0054	0.2549
	CPPG	4.92	0.4272	0.1006	0.3884	4.93	0.4507	0.0695	0.0820	0.3720
	CEPG	6.49	0.4874	0.1485	0.4721	5.04	0.5341	0.1117	0.1438	0.4026
MANU	Unlearned	1.37	0.3656	0.0827	0.0282	1.34	0.3701	0.1101	0.0500	0.2432
	Gaussian	0.17	0.2712	0.0047	0.0053	0.02	0.2213	0.0421	0.0101	0.2198
	Soft Prompt	2.32	0.4728	0.0734	0.1596	4.07	0.5303	0.0691	0.1777	0.5734
	SUA	2.75	0.3712	0.0425	0.0672	2.93	0.4279	0.0759	0.0747	0.2752
	FigStep	0.14	0.2269	0.0059	0.0013	0.21	0.3479	0.0378	0.0342	0.2126
	GCG	0.88	0.4085	0.0872	0.1781	3.68	0.3899	0.0354	0.0645	0.4277
	CPPG	2.93	0.4451	0.0926	0.1877	6.28	0.5796	0.1173	0.1133	0.4675
	CEPG	5.68	0.4938	0.1287	0.2938	7.25	0.5564	0.1678	0.1875	0.6172

Table 1: Recovery attack performance on LLaVA1.5-7B after five multimodal unlearning methods across CLEAR and MLLMU. **Bold/underline** mark the best/runner-up attack within each block; Vanilla denotes the original model before unlearning. Fact. denotes Factuality Score and R-L denotes ROUGE-L. Acc./Choice and Cloze are classification and cloze metrics. Higher values indicate stronger recovery.

Unlearning	CLEAR		MLLMU	
	CPPG	CEPG	CPPG	CEPG
SimNPO	29.06	50.98	38.15	78.91
NPO	32.28	51.66	32.60	64.95
GA	31.76	44.97	31.06	57.08
RMU	36.13	50.16	24.84	39.31
MANU	18.11	35.92	45.29	60.71
Avg.	29.47	46.74	34.28	60.19

Table 2: Recovery ratio (Rec, %) of CPPG and CEPG. Higher values indicate stronger recovery; Avg. averages over the five unlearning methods.

required by GCG. CPPG requires 1.32 GPU-hours. Their median generation latency is 12.36 ms and 23.66 ms, respectively, excluding victim-model inference. ARU takes 7.26 GPU-hours for five epochs; MLLM updates account for 55% of the cost, while generator refitting accounts for 21%. The complete protocol and cost breakdown are reported in Appendix C.

5.4 Analysis of CPPG

CPPG uses CLIP features as transferable semantic conditions, following CLIP-guided generative attacks (Fang et al., 2024). The full model reaches 38.15% Rec and a CLIP distance of 0.188. Removing the image or text branch lowers Rec to 20.83% and 15.67%. Directly projecting CLIP

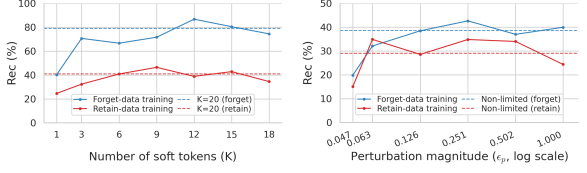
Method	MLLMU				
	Fact.	R-L	BLEU	Cloze	Choice
Qwen2.5-VL-3B-Instruct					
Vanilla	7.17	0.5973	0.3104	0.3080	0.6072
Unlearned	1.16	0.2265	0.0289	0.0012	0.2012
Soft Prompt	5.03	0.4973	0.0775	0.1491	0.3255
SUA	1.62	0.3195	0.0347	0.0703	0.2423
GCG	2.53	0.2498	0.0457	0.1080	0.2554
CPPG	3.88	0.4228	0.0894	0.1526	0.4729
CEPG	6.50	0.5418	0.1001	0.2113	0.4878
InternVL2.5-4B					
Vanilla	5.21	0.4629	0.2991	0.3742	0.4609
Unlearned	0.26	0.1470	0.0363	0.0040	0.1715
Soft Prompt	4.16	0.3976	0.0883	0.1195	0.2950
SUA	2.25	0.1556	0.0814	0.0625	0.2677
GCG	0.34	0.2490	0.0504	0.0393	0.2516
CPPG	3.65	0.3432	0.0903	0.0814	0.3144
CEPG	3.99	0.4041	0.1233	0.1445	0.3438

Table 3: More results of the recovery framework on MLLMU. Metrics and ranking conventions follow Table 1.

features or removing the decoder reduces Rec to 0.89% and -2.67% , with LPIPS above 1.0. These results support joint conditioning, feature adaptation, and spatial decoding. Full results are provided in Appendix D.

5.5 Sensitivity to Key Hyperparameters

We further analyze the number of inserted soft tokens K for CEPG and the pixel perturbation budget ϵ_p for CPPG. Figure 3 reports Rec under forget-data training and retain-only training.



(a) CEPG Rec as the number of soft tokens K varies. (b) CPPG Rec as the perturbation budget ϵ_p varies.

Figure 3: Hyperparameter sensitivity. Dashed lines denote the corresponding default settings.

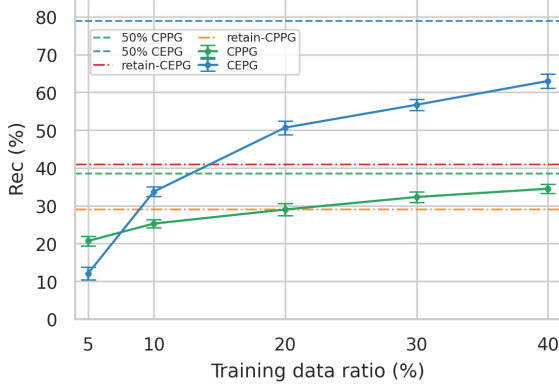
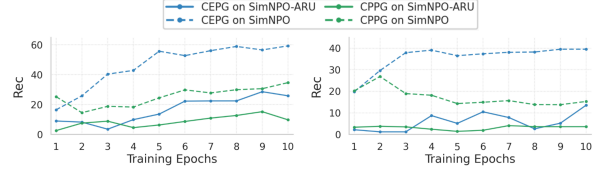


Figure 4: Data efficiency of perturbation generators on SimNPO. Dashed and dash-dotted lines show the 50% and retain-only reference settings.

As shown in Figure 3(a), CEPG does not require a large token budget: recovery rises sharply when K moves beyond very small values, and then remains close to the default setting. CPPG exhibits a similar trend. Increasing ϵ_p from the smallest budgets improves recovery, but larger budgets do not yield a monotonic gain. Thus, the attack is not simply driven by longer soft prompts or larger image changes. The retain-only curves also show measurable recovery, which suggests that the generator can exploit transferable image-text semantics rather than only memorizing exposed forget samples.

5.6 Data Efficiency of the Perturbation Generator

We vary the amount of exposed forget data for SimNPO from 5% to 40%, while using the 50% and retain-only settings as references in Figure 4. CEPG is highly data efficient: its Rec is already non-trivial at 5%, exceeds the retain-only reference after roughly 20% exposure, and continues to improve as more forget samples are available. CPPG increases more gradually, but still recovers meaningful forgotten behavior with limited data. The retain-only baselines further show that direct access



(a) Generators trained on the forget set. (b) Generators trained on the retain set.

Figure 5: Recovery attacks against SimNPO and SimNPO-ARU. Dashed and solid lines denote attacks on SimNPO and SimNPO-ARU, respectively.

to forget samples is not strictly necessary. These results point to a practical risk: user-conditioned perturbation generators can use limited leaked data, or even related retain data, to probe residual forgotten knowledge.

5.7 Evaluation of the ARU

We apply ARU to SimNPO on MLLMU to test whether attack-in-the-loop unlearning can reduce conditional recovery while preserving retained knowledge. Table 4 reports clean forgetting and utility, and Figure 5 reports recovery attacks with generators trained on forget or retain samples.

The results show that ARU improves the forgetting–utility trade-off. As shown in Table 4, SimNPO-ARU further lowers forget-set ROUGE-L, BLEU, and Cloze scores compared with SimNPO. It also improves retain metrics, with only a small drop on Choice. The comparison with SimNPO-ARU* further supports the role of the mask: the masked version preserves stronger retain-set and Real Celebrity performance. Figure 5 shows the same trend under attack. For both forget-set and retain-only generator training, CEPG and CPPG obtain lower Rec against SimNPO-ARU than against SimNPO. These results indicate that ARU reduces conditional recovery while maintaining competitive utility.

ARU is trained only with CEPG. We further evaluate unseen attacks across pixel, embedding, and discrete-text spaces, including black-box transfer and CPPG branch ablations. Table 11 shows that ARU lowers Rec in every tested setting. This includes black-box generators trained on retain data with a surrogate model and no access to target-model queries or gradients.

We also apply ARU to GA and RMU. Table 5 shows that ARU reduces CEPG and CPPG recovery by 35.45 and 18.32 points for GA, and by 10.84 and 10.26 points for RMU. The full results in Ta-

Models	Forget Set				Retain Set				Real Celebrity			
	R-L (↓)	BLEU (↓)	Cloze (↓)	Choice (↓)	R-L (↑)	BLEU (↑)	Cloze (↑)	Choice (↑)	R-L (↑)	BLEU (↑)	Cloze (↑)	Choice (↑)
Vanilla	0.6486	0.3931	0.4025	0.6917	0.6121	0.3900	0.3713	0.6533	0.4790	0.2305	0.1027	0.5304
SimNPO	0.3184	0.0212	0.0840	0.2064	<u>0.5072</u>	<u>0.2188</u>	<u>0.1750</u>	0.4379	<u>0.2258</u>	<u>0.1680</u>	0.0765	<u>0.2763</u>
SimNPO-ARU	0.1449	<u>0.0076</u>	0.0200	0.3264	0.5740	0.3247	0.3040	<u>0.4098</u>	0.3578	0.1783	<u>0.0743</u>	0.4879
SimNPO-ARU*	<u>0.2077</u>	0.0059	<u>0.0625</u>	<u>0.2781</u>	0.3078	0.1367	0.1596	0.3553	0.1747	0.1263	0.0343	0.2199

Table 4: ARU evaluation on MLLMU. Lower forget-set scores indicate stronger forgetting; higher retain-set and Real Celebrity scores indicate better utility. **Bold/underline** mark the best/runner-up among unlearning variants in each column according to the indicated metric direction; ARU* removes the attribution-guided parameter mask.

Objective	CEPG Rec (%)			CPPG Rec (%)		
	Base	ARU	Reduction	Base	ARU	Reduction
GA	57.08	21.63	35.45	31.06	12.74	18.32
RMU	39.31	28.47	10.84	24.84	14.58	10.26

Table 5: Recovery against GA- and RMU-based ARU on MLLMU. Reduction is the base Rec minus the corresponding ARU Rec.

ble 12 show that most utility on the Retain and Real Celebrity sets is preserved. Removing the attribution-guided mask causes larger utility losses for both objectives. MANU is not included because its neuron-pruning update is not compatible with the gradient-based mask used by ARU.

5.8 Evaluation on UMU-Bench

We use a person-disjoint split with 250 forget and 250 retain profiles, keeping paired QA and VQA samples together. Single-modality SimNPO leaves substantial knowledge in the other modality: VQA-only training retains a QA forget score of 0.613, while QA-only training retains a VQA forget score of 0.685. Balanced training reduces the modality gap to 0.027. With the same 1:1 objective, ARU lowers QA/VQA forget scores from 0.277/0.304 to 0.169/0.186 and reduces mean CEPG recovery from 55.7% to 26.3%. It also restores retain accuracy and ROUGE-L to 0.724 and 0.707. CPPG is evaluated separately on VQA because pixel perturbations do not apply to text-only inputs. The complete results and metric definitions are provided in Appendix E.

6 Conclusion

In this paper, we introduced a conditional perturbation generator attack framework to evaluate the recovery risks of unlearning methods applied to MLLMs. The framework generates input-conditioned perturbations for an unlearned MLLM, rather than relying on fixed prompts or universal perturbations. Experiments on MLLMU-Bench, CLEAR, and UMU-Bench show that such con-

ditional attacks can recover forgotten knowledge across multiple unlearning methods, even with limited or no access to forgotten samples, providing a more effective assessment of recovery risk. We further proposed ARU, which incorporates conditional recovery attacks into unlearning. Its attribution-guided mask selects a more suitable parameter update subspace to balance forgetting effectiveness and utility preservation, reducing recovery under conditional attacks while maintaining competitive retain-set performance.

Limitations

We study recovery under access to the unlearned model. In a pure black-box setting, the attacker only observes final text outputs, without gradients, internal features, or the embedding interface. Extending conditional perturbation generation to this setting requires stronger transferability. ARU is also limited to differentiable unlearning objectives, since its attribution-guided mask depends on gradients. It may hard apply to pruning-based or other non-gradient methods. Although ARU reduces conditional recovery, partial recovery can still occur, indicating that stronger MLLM unlearning methods are needed to more reliably remove textual, visual, and cross-modal memories.

Ethical Considerations

This work examines recovery risks in MLLM unlearning, which may be useful for auditing but could also be misused to elicit forgotten or sensitive information. The proposed attacks could be misused to elicit forgotten or sensitive information, but our study is intended for controlled auditing and defense development. All experiments are conducted on offline benchmarks rather than deployed systems, and results are reported in aggregate form. We encourage the use of these methods only for authorized evaluation of unlearning robustness and for developing stronger defenses.

References

- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Huihui Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. [Qwen2.5-vl technical report](#). *Preprint*, arXiv:2502.13923.
- Lucas Bourtole, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. [Machine unlearning](#). In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159.
- Yinzhi Cao and Junfeng Yang. 2015. [Towards making systems forget with machine unlearning](#). In *2015 IEEE Symposium on Security and Privacy*, pages 463–480.
- Haokun Chen, Sebastian Szlyler, Weilin Xu, and Nageen Himayat. 2025. Soft token attacks cannot reliably audit unlearning in large language models. *arXiv preprint arXiv:2502.15836*.
- Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.
- Jiali Cheng and Hadi Amiri. 2024. Multidelete for multimodal machine unlearning. In *European Conference on Computer Vision*, pages 165–184. Springer.
- Alexey Dontsov, Dmitrii Korzh, Alexey Zhavoronkin, Boris Mikheev, Denis Bobkov, Aibek Alanov, Oleg Rogov, Ivan Oseledets, and Elena Tutubalina. 2025. [CLEAR: Character unlearning in textual and visual modalities](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20582–20603, Vienna, Austria. Association for Computational Linguistics.
- Jai Doshi and Asa Cooper Stickland. 2025. [Does unlearning truly unlearn? a black box evaluation of llm unlearning methods](#). *Preprint*, arXiv:2411.12103.
- Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. 2025. [Simplicity prevails: Rethinking negative preference optimization for llm unlearning](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 1540–1567. Curran Associates, Inc.
- Hao Fang, Jiawei Kong, Bin Chen, Tao Dai, Hao Wu, and Shu-Tao Xia. 2024. Clip-guided generative networks for transferable targeted adversarial attacks. In *European Conference on Computer Vision*, pages 1–19. Springer.
- Weiwei Feng, Nanqing Xu, Tianzhu Zhang, and Yongdong Zhang. 2023. Dynamic generative targeted attacks with pattern injection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16404–16414.
- GDPR. 2016. [Regulation \(EU\) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC \(General Data Protection Regulation\)](#). Official Journal of the European Union. OJ L 119, 4.5.2016, pp. 1–88.
- Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. 2025. Figstep: Jailbreaking large vision-language models via typographic visual prompts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23951–23959.
- Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. 2020. [Certified data removal from machine learning models](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3832–3842. PMLR.
- Shengyuan Hu, Yiwei Fu, Steven Wu, and Virginia Smith. 2025. [Unlearning or obfuscating? jogging the memory of unlearned llms via benign relearning](#). In *International Conference on Learning Representations*, volume 2025, pages 8857–8888.
- Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. 2024. [Catastrophic jailbreak of open-source llms via exploiting generation](#). In *International Conference on Learning Representations*, volume 2024, pages 13707–13727.
- Jiahao Huo, Yibo Yan, Xu Zheng, Yuanhuiyi Lyu, Xin Zou, Zhihua Wei, and Xuming Hu. 2025. [MMUnlearner: Reformulating multimodal machine unlearning in the era of multimodal large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7190–7206, Vienna, Austria. Association for Computational Linguistics.
- Jinghan Jia, Jiancheng Liu, Yihua Zhang, Parikshit Ram, Nathalie Baracaldo, and Sijia Liu. 2024. [Wagle: Strategic weight attribution for effective and modular unlearning in large language models](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 55620–55646. Curran Associates, Inc.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jiaqi Li, Qianshan Wei, Chuanyi Zhang, Guilin Qi, Miaozen Du, Yongrui Chen, Sheng Bi, and Fan Liu. 2024a. [Single image unlearning: Efficient machine unlearning in multimodal large language models](#). In

- Advances in Neural Information Processing Systems*, volume 37, pages 35414–35453. Curran Associates, Inc.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.
- Miles Q. Li and Benjamin C.M. Fung. 2025. [Security concerns for large language models: A survey](#). *Journal of Information Security and Applications*, 95:104284.
- Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew Bo Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, and 27 others. 2024b. [The WMDP benchmark: Measuring and reducing malicious use with unlearning](#). In *Forty-first International Conference on Machine Learning*.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Fan Liu, Zhao Xu, and Hao Liu. 2024a. Adversarial tuning: Defending against jailbreak attacks for llms. *arXiv preprint arXiv:2406.06622*.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024b. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.
- Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024c. [Autodan: Generating stealthy jailbreak prompts on aligned large language models](#). In *International Conference on Learning Representations*, volume 2024, pages 56174–56194.
- Xuannan Liu, Xing Cui, Peipei Li, Zekun Li, Huaibo Huang, Shuhan Xia, Miaoxuan Zhang, Yueying Zou, and Ran He. 2024d. [Jailbreak attacks and defenses against multimodal generative models: A survey](#). *Preprint*, arXiv:2411.09259.
- Zheyuan Liu, Guangyao Dou, Mengzhao Jia, Zhaoxuan Tan, Qingkai Zeng, Yongle Yuan, and Meng Jiang. 2025a. [Protecting privacy in multimodal large language models with MLLMU-bench](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4105–4135, Albuquerque, New Mexico. Association for Computational Linguistics.
- Zheyuan Liu, Guangyao Dou, Xiangchi Yuan, Chunhui Zhang, Zhaoxuan Tan, and Meng Jiang. 2025b. [Modality-aware neuron pruning for unlearning in multimodal large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5913–5933, Vienna, Austria. Association for Computational Linguistics.
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. [Towards deep learning models resistant to adversarial attacks](#). In *International Conference on Learning Representations*.
- Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. 2025. [TOFU: A task of fictitious unlearning for LLMs](#). In *Red Teaming GenAI: What Can We Learn from Adversaries?*
- Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. 2021. On generating transferable targeted perturbations. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7708–7717.
- Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2025. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72.
- OpenAI. 2025. Gpt-5 system card. <https://openai.com/index/gpt-5-system-card/>. Accessed: 2026-05-12.
- Shilong Pan, Zhiliang Tian, Zhen Huang, Wanlong Yu, Zhihua Wen, Xinwang Liu, Kai Lu, Minlie Huang, and Dongsheng Li. 2025. [AGD: Adversarial game defense against jailbreak attacks in large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17391–17406, Vienna, Austria. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. 2018. Generative adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4422–4431.

- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Leo Schwinn, David Dobre, Sophie Xhonneux, Gauthier Gidel, and Stephan Günnemann. 2024. [Soft prompt threats: Attacking safety alignment and unlearning in open-source llms through the embedding space](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 9086–9116. Curran Associates, Inc.
- Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah Smith, and Chiyuan Zhang. 2025. [Muse: Machine unlearning six-way evaluation for language models](#). In *International Conference on Learning Representations*, volume 2025, pages 27797–27818.
- Chengye Wang, Yuyuan Li, Xiaohua Feng, Chaochao Chen, Xiaolin Zheng, and Jianwei Yin. 2025. [UMU-Bench: Closing the modality gap in multimodal unlearning evaluation](#). In *Advances in Neural Information Processing Systems*.
- Xiao Yang, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2022. Boosting transferability of targeted adversarial examples via hierarchical generative networks. In *European conference on computer vision*, pages 725–742. Springer.
- Yuanshun Yao, Xiaojun Xu, and YangLiu. 2024. [Large language model unlearning](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 105425–105475. Curran Associates, Inc.
- Hongbang Yuan, Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2025. Towards robust knowledge unlearning: An adversarial framework for assessing and improving unlearning robustness in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25769–25777.
- Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024. [Negative preference optimization: From catastrophic collapse to effective unlearning](#). In *First Conference on Language Modeling*.
- Xianren Zhang, Hui Liu, Delvin Ce Zhang, Xianfeng Tang, Qi He, Dongwon Lee, and Suhan Wang. 2025. [SUA: Stealthy multimodal large language model unlearning attack](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11230–11243, Suzhou, China. Association for Computational Linguistics.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhaghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.
- Andy Zhou, Bo Li, and Haohan Wang. 2024. [Robust prompt optimization for defending language models against jailbreaking attacks](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 40184–40211. Curran Associates, Inc.
- Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. [Universal and transferable adversarial attacks on aligned language models](#). *Preprint*, arXiv:2307.15043.

A Unlearning Methods

We use the notation in Section 3. For a multimodal sample (I, x, y) , let

$$\ell_{\text{NLL}}(\theta; I, x, y) = - \sum_{t=1}^{|y|} \log p_{\theta}(y_t \mid H(I, x), y_{<t}), \quad (11)$$

where $H(I, x)$ is the multimodal context of M_{θ} . The pre-unlearning model is M_{θ_o} , and the retain term is the NLL on $\mathcal{D}_r$ unless stated otherwise.

Gradient ascent (GA). GA makes the target response unlikely by minimizing the negative likelihood on $\mathcal{D}_f$:

$$\mathcal{L}_{\text{GA}} = \mathbb{E}_{\mathcal{D}_f} \log p_{\theta}(y_f \mid I, x) + \lambda_r \mathbb{E}_{\mathcal{D}_r} \ell_{\text{NLL}}(\theta; I_r, x_r, y_r). \quad (12)$$

This simple update can suppress neighboring knowledge because it lacks preference saturation (Yao et al., 2024).

NPO and SimNPO. NPO treats y_f as a negative response relative to M_{θ_o} . With $\beta > 0$,

$$\begin{aligned} \ell_f^{\text{NPO}} &= -\frac{2}{\beta} \log \sigma(r_{\theta}^{\text{NPO}}), \\ r_{\theta}^{\text{NPO}} &= -\beta \log \frac{p_{\theta}}{p_{\theta_o}}(y_f \mid I, x). \end{aligned} \quad (13)$$

SimNPO uses a length-normalized score

$$\begin{aligned} \ell_f^{\text{SimNPO}} &= -\frac{2}{\beta} \log \sigma(r_{\theta}^{\text{Sim}}), \\ r_{\theta}^{\text{Sim}} &= -\frac{\beta}{|y_f|} \log p_{\theta}(y_f \mid I, x) - \gamma. \end{aligned} \quad (14)$$

The sigmoid weakens updates after the target response becomes unlikely, while the SimNPO normalization reduces response-length bias (Zhang et al., 2024; Fan et al., 2025).

RMU. RMU modifies layer- ℓ hidden states. Let $m_f(t)$ and $m_r(t)$ be activations on forget and retain contexts, $m_o(t)$ the matching activation of M_{θ_o} , and u a fixed random unit vector. RMU uses

$$\begin{aligned} \mathcal{L}_{\text{RMU}} &= \mathcal{L}_{\text{for}} + \alpha \mathcal{L}_{\text{ret}}, \\ \mathcal{L}_{\text{for}} &= \mathbb{E}_{\mathcal{D}_f} L_f^{-1} \sum_t \|m_f(t) - cu\|_2^2, \\ \mathcal{L}_{\text{ret}} &= \mathbb{E}_{\mathcal{D}_r} L_r^{-1} \sum_t \|m_r(t) - m_o(t)\|_2^2. \end{aligned} \quad (15)$$

The first term misdirects forget activations, and the second preserves retain representations (Li et al., 2024b).

MANU. MANU targets MLLMs by first identifying neurons that contribute to forget knowledge under textual, visual, and cross-modal inputs, and then clipping or pruning the selected neurons. We use MANU as an evaluated pruning-based unlearning method, not as the differentiable base objective inside ARU (Liu et al., 2025b).

B Recovery Baselines

Gaussian noise is a non-adaptive image perturbation. The standard deviation is set to 12/255. GCG searches for a discrete suffix z over the vocabulary $\mathcal{V}$:

$$\begin{aligned} z^* &= \arg \min_{z \in \mathcal{V}^T} - \sum_{t=1}^{|y_f|} \log p_{\theta_u}(y_{f,t} \mid C_z, y_{f,<t}), \\ C_z &= H(I, [x; z]). \end{aligned} \quad (16)$$

using gradient-ranked token replacement and forward evaluation (Zou et al., 2023). FigStep moves part of the elicitation signal to the visual channel by rendering text as a typographic prompt (Gong et al., 2025).

Soft Token Attack optimizes continuous prompt embeddings for the frozen M_{θ_u} (Schwinn et al., 2024). We use its universal setting:

$$\begin{aligned} \Delta^{S^*} &= \arg \min_{\|\Delta^S\| \leq c} \mathbb{E}_{\mathcal{D}_f^{\text{atk}}} [\ell_S(\Delta^S)], \\ \ell_S(\Delta^S) &= - \sum_{t=1}^{|y_f|} \log p_{\theta_u}(y_{f,t} \mid C^S, y_{f,<t}). \end{aligned} \quad (17)$$

where C^S inserts the shared soft prompt at the same position as CEPG. We do not use the individual setting, which optimizes a new soft prompt for each test input and its own reference answer. Such per-test optimization can inject target supervision into the audit and force surface-string reproduction rather than measure residual recovery. Recent analysis also questions the reliability of individual soft-token audits (Chen et al., 2025). The universal setting is a stricter transfer test and matches CEPG in split, insertion position, token length, norm bound, and schedule.

SUA learns a universal visual perturbation for MLLM unlearning attacks. It optimizes target recovery with a semantic alignment term, so perturbed and denoised image features remain close (Zhang et al., 2025). In our experiments, SUA uses the same forget-data split, optimization schedule,

clipping, and normalization as CPPG. Table 6 summarizes the hyperparameter settings used in our experiments.

(a) Training and data settings				
Method	Schedule	Batch	Split	Leaked data
CEPG	5 epochs	2	250 / 250	50% (125 / 250)
CPPG	20 epochs	2	250 / 250	50% (125 / 250)
GCG	4000 steps	6	250 / 250	50% (125 / 250)
SUA	4000 steps	6	250 / 250	50% (125 / 250)
Soft Prompt	5 epochs	6	250 / 250	50% (125 / 250)

(b) Optimization settings					
Method	Optimizer / update	LR / step	Warm.	Length	Budget
CEPG	AdamW	1×10^{-5}	0.03	20	$\ z_i\ _2 \leq 1.5$ $ z_i \leq 0.2$
CPPG	AdamW	1×10^{-4}	0.03	—	—
GCG	Search, top- $k=16$ width = 16	—	—	20	—
SUA	Sign gradient	1/255	—	—	—
Soft Prompt	Sign gradient	1×10^{-3}	—	20	$\ z_i\ _2 \leq 1.5$ $ z_i \leq 0.2$

Table 6: Hyperparameter settings for the perturbation generators and baseline attacks on MLLMU-Bench. The split is reported at the profile level, and leaked data denotes the exposed fraction of forget profiles.

C Computational Efficiency

For the cost comparison, each attack uses 11,250 sample exposures. GCG, Soft Prompt, and SUA run 1,875 updates with batch size 6. CEPG and CPPG make five passes over 2,250 samples. We measure training time on two RTX 4090 GPUs and exclude evaluation. GCG evaluates 16 candidates at each update.

Method	Exp.	GPU-h	Rec. (%) $\uparrow$	Latency (ms) Median / P95
GCG	11,250	7.92	4.32	0.025 / 0.026
Soft Prompt	11,250	1.06	30.59	0.028 / 0.028
SUA	11,250	1.10	7.68	0.222 / 0.229
CEPG	11,250	1.24	78.91	12.36 / 16.63
CPPG	11,250	1.32	18.24	23.66 / 24.00

Table 7: Attack cost under matched sample exposure. Latency uses batch size 1 and excludes victim-model inference.

Latency is measured after 20 warm-up runs over 200 synchronized trials. For the baselines, it includes applying a cached perturbation. For CEPG and CPPG, it includes one generator pass and perturbation application.

ARU component	Workload	GPU-h	Share
MLLM updates	1 + 4 epochs	3.99	55.0%
Generator refits	3×2 epochs	1.53	21.0%
Attribution scoring	4 full passes	1.74	24.0%
Total	5 epochs	7.26	100%

Table 8: Cost breakdown of ARU.

MLLM updates account for most of the ARU cost. Generator refitting takes 21%, so it is not the main bottleneck.

D CPPG Ablation

CPPG uses frozen CLIP features as semantic conditions, following CLIP-guided generative attacks (Fang et al., 2024). Control tokens provide a compact condition, the fusion module combines it with spatial image features, and the decoder maps the fused features to a pixel perturbation. All variants use the same training settings.

Variant	Rec. (%) $\uparrow$	LPIPS $\downarrow$	CLIP-dist $\downarrow$
CPPG	38.15	<u>0.431</u>	0.188
w/o image branch	<u>20.83</u>	0.454	<u>0.384</u>
w/o text branch	15.67	0.357	0.543
Direct CLIP	0.89	1.001	0.488
w/o decoder	-2.67	1.041	0.508
Pooled CLIP + FiLM	17.74	0.527	0.772

Table 9: CPPG ablation on MLLMU. LPIPS uses the AlexNet backbone. Bold and underline mark the best and runner-up variants for each metric.

The full model gives the strongest recovery and the lowest CLIP distance. Removing either condition branch weakens recovery. Direct CLIP projection and removing the decoder also produce the largest LPIPS values. The text-free variant has lower LPIPS but weaker recovery and semantic alignment, showing that pixel similarity alone is insufficient.

E Evaluation on UMU-Bench

UMU-Bench (Wang et al., 2025) evaluates aligned forgetting across text-only QA and image-text VQA. We use a person-disjoint 1:1 split with 250 forget and 250 retain profiles. Paired QA and VQA samples remain in the same split. The training objective is $L = \alpha L_{VQA} + \beta L_{QA}$, where $\alpha : \beta$ gives the VQA-to-QA weight ratio.

VQA-only or QA-only training leaves high forget scores in the untargeted modality. Balanced SimNPO reduces the modality gap to 0.027. Balanced ARU keeps this gap at 0.028, lowers both forget scores, and reduces mean CEPG recovery from 55.7% to 26.3%. It also restores retain utility close to the original model. CPPG is evaluated separately on VQA because pixel perturbations do not apply to text-only QA.

(a) SimNPO				
Metric	Origin	1:0	1:1	0:1
QA/VQA-F ↓	0.728/0.866	0.613/ 0.238	<u>0.277/0.304</u>	0.196/0.685
Δ_{mod} ↓	0.138	<u>0.375</u>	0.027	0.489
AccF/RL-F ↓	0.837/0.872	0.290/0.432	<u>0.291/0.320</u>	0.408/ 0.294
RecQ/RecV ↓	–	28.7/65.2	<u>51.8/59.6</u>	63.7/ 30.9
AccR/RL-R ↑	0.743/0.731	0.522/0.605	<u>0.409/0.592</u>	<u>0.516/0.498</u>

(b) SimNPO-ARU		
Metric	1:0	1:1
QA/VQA-F ↓	<u>0.293/0.172</u>	0.169/0.186
Δ_{mod} ↓	<u>0.121</u>	0.028
AccF/RL-F ↓	<u>0.291/0.339</u>	0.254/0.207
RecQ/RecV ↓	22.4/27.8	<u>28.9/23.6</u>
AccR/RL-R ↑	<u>0.630/0.712</u>	0.724/0.707

Table 10: Unlearning and recovery on UMU-Bench. Columns give $\alpha : \beta$. QA/VQA-F aggregates classification, cloze, and generation ROUGE-L on the forget set. Δ_{mod} is the absolute modality gap. RecQ/RecV is CEPG recovery on QA/VQA. AccR/RL-R reports retain accuracy and ROUGE-L. Bold and underline mark the best and runner-up unlearning settings for each metric component; Origin is not included in the ranking.

F Evaluation Metrics

For open-ended responses, we report ROUGE-L recall and BLEU. Let $\hat{y}$ be an answer generated by the model and y the reference answer. ROUGE-L recall measures how much of the reference is covered by the longest common subsequence:

$$\text{R-L}_{\text{recall}} = \frac{\text{LCS}(\hat{y}, y)}{|y|}. \quad (18)$$

It is useful for privacy-style QA because missing an identity attribute lowers recall even if the answer is fluent. BLEU reports clipped n -gram precision with a brevity penalty,

$$\text{BLEU} = \text{BP} \exp \left(\sum_{n=1}^N w_n \log p_n \right), \quad (19)$$

where p_n is the clipped precision for n -grams. It complements ROUGE-L by penalizing unsupported surface overlap and word-order mismatches (Lin, 2004; Papineni et al., 2002). For factuality, we use an LLM judge with the following fixed prompt template (Zheng et al., 2023). We repeat the scoring five times for each answer and report the average score to reduce evaluation variance.

Factuality Evaluation Prompt

You are an expert factuality judge for multimodal QA.

- Assign a **Factuality Score** between 1 and 10 comparing the candidate answer to the reference.
- 10-9: fully aligned in meaning; 8-7: minor omissions; 6-5: partial correctness; 4-3: major errors; 2-1: incorrect or irrelevant.
- Keep the justification short (1-2 sentences).

Example JSON:

```
{
  "Factuality Score": 9,
  "Justification": "Matches all identity attributes except minor detail."
}
```

Image ID: {image_id}
 Question: {question}
 Candidate: {prediction}
 Reference: {reference}

G Weight Attribution Score

For $\theta_i \in \mathcal{C}$, let $\bar{g}_i^c$, $\bar{g}_i^e$, and $\bar{g}_i^f$ be the B -batch average gradients of the clean-forget, embedding-perturbed forget, and retain losses. ARU uses the robust forget gradient

$$\bar{g}_i^f = (1 - \alpha)\bar{g}_i^c + \alpha\bar{g}_i^e. \quad (20)$$

We derive the mask score from the WAGLE sensitivity view (Jia et al., 2024). Introduce a multiplicative gate ϵ and define the retain-constrained weights

$$\begin{aligned} \theta^*(\epsilon) &= \arg \min_{\theta} \mathcal{L}_r(\epsilon \odot \theta), \\ S(\epsilon) &= \mathcal{L}_f(\epsilon \odot \theta^*(\epsilon)) - \mathcal{L}_f(\theta^*(\mathbf{1})). \end{aligned} \quad (21)$$

Here $\mathcal{L}_f$ is the robust forget loss whose gradient is $\bar{g}_i^f$, and $\mathcal{L}_r$ is the retain loss. Let $z(\epsilon) = \epsilon \odot \theta^*(\epsilon)$. A first-order expansion at $\epsilon = \mathbf{1}$ gives

$$S(\epsilon) \approx (\epsilon - \mathbf{1})^\top \left. \frac{\partial z(\epsilon)}{\partial \epsilon} \right|_{\epsilon=\mathbf{1}} \nabla_{\theta} \mathcal{L}_f. \quad (22)$$

The stationarity condition of the lower level, combined with implicit differentiation, gives

$$\begin{aligned} \frac{\partial \theta^*}{\partial \epsilon} &= -\nabla_{\epsilon, \theta} \mathcal{L}_r (\nabla_{\theta, \theta} \mathcal{L}_r)^{-1} \\ &\approx -\lambda_{\text{attr}} \text{diag}(\nabla_{\theta} \mathcal{L}_r), \end{aligned} \quad (23)$$

where the last step uses the diagonal Hessian approximation and absorbs the inverse Hessian scale into λ_{attr} . Therefore,

$$S(\epsilon) \approx (\epsilon - \mathbf{1})^\top [(\theta - \lambda_{\text{attr}} \epsilon \odot \nabla_{\theta} \mathcal{L}_r) \odot \nabla_{\theta} \mathcal{L}_f]. \quad (24)$$

For a single coordinate gate $\epsilon = \mathbf{1} + \mu e_i$ with small $|\mu|$,

$$S_i \approx \mu(\theta_i \bar{g}_i^f - \lambda_{\text{attr}} \bar{g}_i^r \bar{g}_i^f) - \mu^2 \lambda_{\text{attr}} \bar{g}_i^r \bar{g}_i^f. \quad (25)$$

The $O(\mu^2)$ term is discarded because the attribution is local around $\epsilon = 1$. Since the mask ranks coordinates and does not use the sign of μ , ARU scores each weight by the magnitude

$$A_i = \left| \theta_i \bar{g}_i^f - \lambda_{\text{attr}} \bar{g}_i^f \bar{g}_i^r \right|. \quad (26)$$

The first term is the direct first-order effect of θ_i on robust forgetting. The second term discounts weights whose forget gradient is strongly coupled with the retain gradient. ARU keeps the top- ρ fraction of A_i and applies the base update only in this subspace.

H Additional ARU Evaluations

H.1 Transfer to Unseen Recovery Attacks

ARU uses CEPG as the recovery attack during training. We evaluate the resulting SimNPO-ARU model with attacks that are not used in this process. These tests cover pixel, embedding, and discrete-text perturbations, as well as different access assumptions and CPPG branch ablations.

Attack	Space	SimNPO	+ARU	Reduction
SUA	Pixel	12.48	9.04	3.44
SUA (grey-box)	Pixel	7.13	3.89	3.24
Soft Prompt	Embedding	26.45	11.06	15.39
GCG	Discrete text	6.94	6.56	0.38
CEPG (black-box)	Embedding	24.14	16.33	7.81
CPPG (black-box)	Pixel	17.23	14.47	2.76
CPPG (non-image branch)	Pixel	20.83	15.09	5.74
CPPG (non-text branch)	Pixel	15.67	7.44	8.23

Table 11: Recovery ratio (Rec, %) of attacks not used to train ARU on MLLMU. Reduction is the SimNPO score minus the SimNPO-ARU score.

ARU lowers the point-estimate Rec for every tested attack. The black-box generators are trained only on the retain set with a surrogate that has the same initialization and unlearning method. They use no target-model queries or gradients. The smaller absolute changes for SUA and GCG coincide with their lower initial recovery ratios.

H.2 Transfer Across Unlearning Objectives

We apply ARU to GA and RMU to cover gradient-ascent and representation-based unlearning objectives. ARU keeps the corresponding base objective for clean and adversarial forget inputs. Table 12 reports the final results after 10 defense epochs.

(a) GA-based ARU					
Set	Metric	Vanilla	GA	+ARU	+ARU*
Forget	R-L ↓	0.6486	0.1256	0.0867	0.0749
	BLEU ↓	0.3931	0.0024	0.0030	0.0017
	Cloze ↓	0.4025	0.0040	0.0031	0.0053
	Choice ↓	0.6917	0.1264	0.1758	<u>0.1516</u>
Retain	R-L ↑	0.6121	0.3847	0.4284	0.2719
	BLEU ↑	0.3900	0.1362	0.1876	0.1084
	Cloze ↑	0.3713	0.1215	0.1668	0.0973
	Choice ↑	0.6533	0.3378	<u>0.2815</u>	0.2746
Real	R-L ↑	0.4790	0.1546	0.3067	0.1428
	BLEU ↑	0.2305	0.0718	0.0948	0.0741
	Cloze ↑	0.1027	0.0439	<u>0.0406</u>	0.0257
	Choice ↑	0.5304	<u>0.2265</u>	0.2889	0.1698
(b) RMU-based ARU					
Set	Metric	Vanilla	RMU	+ARU	+ARU*
Forget	R-L ↓	0.6486	0.4134	0.2183	0.3746
	BLEU ↓	0.3931	0.0509	<u>0.0176</u>	0.0138
	Cloze ↓	0.4025	0.0120	0.0068	<u>0.0095</u>
	Choice ↓	0.6917	0.2648	0.3174	<u>0.2926</u>
Retain	R-L ↑	0.6121	0.5486	0.5092	0.4581
	BLEU ↑	0.3900	0.3027	0.3316	0.2475
	Cloze ↑	0.3713	<u>0.2864</u>	0.3187	0.2559
	Choice ↑	0.6533	<u>0.4018</u>	0.4869	0.3296
Real	R-L ↑	0.4790	0.2421	0.3914	0.2784
	BLEU ↑	0.2305	0.1856	0.1962	0.1492
	Cloze ↑	0.1027	0.0834	0.0809	0.0737
	Choice ↑	0.5304	0.4217	0.4728	<u>0.4286</u>

Table 12: GA- and RMU-based ARU on MLLMU after 10 defense epochs. Lower forget scores indicate stronger forgetting; higher retain and Real Celebrity scores indicate better utility. Bold and underline mark the best and runner-up variants within each panel. ARU* removes the attribution-guided mask.