MLLMEraser: Achieving Test-Time Unlearning in Multimodal Large Language Models through Activation Steering

Chenlu Ding[1] **Jiancan Wu**[1,*] **Leheng Sheng**[2]
Fan Zhang[3] **Yancheng Yuan**[4,*] **Xiang Wang**[1] **Xiangnan He**[1,*]
[1]University of Science and Technology of China
[2]National University of Singapore
[3]Huawei
[4]The Hong Kong Polytechnic University
dingchenlu200103@gmail.com

Abstract

Multimodal large language models (MLLMs) have demonstrated remarkable capabilities across vision–language tasks, yet their large-scale deployment raises pressing concerns about memorized private data, outdated knowledge, and harmful content. Existing unlearning approaches for MLLMs typically adapt training-based strategies such as gradient ascent or preference optimization, but these methods are computationally expensive, irreversible, and often distort retained knowledge. In this work, we propose MLLMERASER, an input-aware, training-free framework for test-time unlearning. Our approach leverages activation steering to enable dynamic knowledge erasure without parameter updates. Specifically, we construct a multimodal erasure direction by contrasting adversarially perturbed, knowledge-recall image–text pairs with knowledge-erasure counterparts, capturing both textual and visual discrepancies. To prevent unnecessary interference, we further design an input-aware steering mechanism that adaptively determines when and how the erasure direction should be applied, preserving utility on retained knowledge while enforcing forgetting on designated content. Experiments on LLaVA-1.5 and Qwen-2.5-VL demonstrate that MLLMERASER consistently outperforms state-of-the-art MLLM unlearning baselines, achieving stronger forgetting performance with lower computational cost and minimal utility degradation.

1 Introduction

Multimodal large language models (MLLMs) [1–5] have demonstrated remarkable capabilities in tasks such as visual question answering [6, 7], image–text generation [8, 9], and embodied AI applications [8, 10]. However, their large-scale deployment raises concerns about memorizing problematic information once learned, particularly in privacy-sensitive [11] or safety-critical [12] applications, highlighting the need for reliable unlearning mechanisms to ensure trustworthy MLLM systems [13–15]. MLLM unlearning aims to selectively erase designated information across modalities while preserving general utility, thereby supporting privacy protection, mitigating misuse, and maintaining reliability [16, 17]. Existing work mainly adapts training–based strategies from LLM unlearning, employing gradient ascent, preference optimization [18], or performing targeted parameter updates [19, 11]. While effective, these training-based methods introduce substantial computational costs, inference latency, and risks of corrupting retained knowledge [20, 21]. This motivates test-time

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

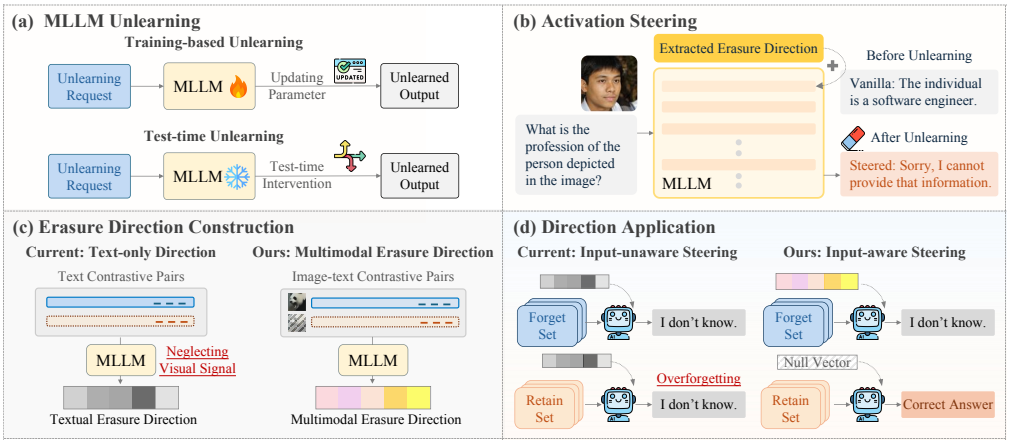


Figure 1: (a) Comparison between training-based and test-time unlearning paradigms for MLLMs. (b) Illustration of the activation steering process. (c)–(d) Differences between existing methods and ours in constructing and applying the steering vector.

MLLM unlearning (See Figure 1), a paradigm that prevents the generation of designated information at inference without modifying model parameters, which offers an immediate, lightweight, and reversible solution.

Recently, activation steering [22, 23] emerges as a promising approach for test-time intervention. Activation steering manipulates the internal computation of LLMs by injecting a carefully constructed direction vector into their intermediate activations, shifting the model's latent representation toward a desired semantic space and inducing specific behaviors or responses. However, existing studies on steering have focused mainly on safety alignment [24, 25], reasoning length regulation [26, 27], and hallucination reduction [28, 29], leaving its potential for test-time MLLM unlearning largely unexplored. Here we aim to leverage the superiority of activation steering for test-time MLLM unlearning, yet encounter two fundamental challenges: multimodal erasure direction construction and multimodal erasure direction application.

- **Multimodal erasure direction construction:** Multimodal erasure direction construction denotes how to construct an effective activation steering vector for MLLMs. Traditional steering methods can directly elicit contrastive activation pairs from existing models by prompting for different response types, such as truthful versus deceptive answers [29] or safe versus unsafe outputs [30]. However, in the context of MLLM unlearning, the ideal contrastive samples would come from models that have already forgotten the target information versus those that retain it—but such "unlearned" models are precisely what we seek to avoid obtaining through expensive retraining. Moreover, given that MLLMs inherently encode joint visual-textual representations through deep cross-modal fusion [1, 31], existing steering works that predominantly rely on textual contrasts [32] while neglecting visual signal processing produce incomplete erasure directions, leading to insufficient forgetting performance.
- **Multimodal erasure direction application:** Multimodal erasure direction application focuses on when to selectively apply this direction. Even with a well-extracted multimodal erasure direction, deciding when to apply it remains an open challenge. Current steering methods typically apply uniform interventions across all inputs [22, 30], which can effectively mitigate unsafe or privacy-leaking outputs but frequently distorts responses to non-targeted queries and degrades performance on retained knowledge [33, 25]. An effective unlearning approach requires input-aware activation that selectively triggers interventions only for content requiring unlearning (*i.e.,* forget set) while preserving normal model behavior for retained information (*i.e.,* retain set). However, unlike scenarios with clear semantic distinctions (*e.g.,* positive vs. negative sentiment), forget and retain examples often share identical formats and content types—both may involve user-attribute queries that differ only in their unlearning designation [14]. Such similarity makes accurate selective control difficult and increases the risk of over-forgetting [34].

To address these challenges, we introduce MLLMEraser, an input-aware test-time MLLM unlearning framework that leverages activation steering for dynamic and test-time information erasure without parameter modification. For the multimodal erasure direction construction, we ground the response-

intervention objective of unlearning in the model's intrinsic refusal behavior and derive the erasure direction by contrasting the semantics of knowledge-recall and knowledge-erasure. Specifically, we generate two contrastive sets to extract the direction: a negative set (*i.e.,* knowledge-recall inputs) combining jailbreak prompts with adversarially perturbed images to induce unsafe or privacy-sensitive outputs, and a positive set (*i.e.,* knowledge-erasure inputs) that elicits refusal-style responses (*e.g.,* "I cannot answer this question.") with the corresponding clean images. After obtaining the steering vector, we reformulate steering as an input-aware task by introducing a direction-determining function $f(\cdot)$, rather than applying the direction indiscriminately. Given the hidden activations $\mathbf{h}$, this function adaptively decides the steering vector $f(\mathbf{h})$. For forget data, $f(\mathbf{h})$ maps the activations towards the pre-computed erasure direction, while for retain data, $f(\mathbf{h})$ degenerates into a null direction [35], yielding nearly zero intervention and leaving the representation distribution unchanged. Inspired by [24], we implement $f(\mathbf{h})$ as a simple yet effective linear transformation, circumventing the need for additional auxiliary model training. Experiments on LLaVA-1.5 [1] and Qwen-2.5-VL [31] demonstrate the effectiveness of MLLMERASER, which consistently outperforms state-of-the-art MLLM unlearning methods while providing an efficient and lightweight solution that balances unlearning performance with model utility preservation. Our code is available at https://github.com/oceanoceanna/MLLMEraser_2026.

2 Preliminary

This section begins by introducing the notation and formalizing the problem of test-time MLLM unlearning in Section 2.1. Section 2.2 outlines how activation steering offers a feasible mechanism for test-time MLLM unlearning, enabling test-time behavior control.

2.1 Notation and Problem Setup

Given a multimodal instruction-tuning dataset $\mathcal{D} = \{(\mathcal{I}_i, \mathcal{Q}_i, \mathcal{A}_i)\}_{i=1}^{N}$ of size N, where $\mathcal{I}_i$ denotes the input image, $\mathcal{Q}_i$ is the textual instruction, and $\mathcal{A}_i = (y_1^{(i)}, y_2^{(i)}, \ldots, y_{|\mathcal{A}_i|}^{(i)})$ represents the target answer sequence, the MLLM is fine-tuned to maximize the likelihood of predicting each token y_t given the multimodal context and the previously generated tokens. Specifically, the image is first encoded by a vision encoder and projected into the language space through a multimodal adapter. This fused representation is then concatenated with the tokenized query and processed autoregressively by the LLM backbone to generate the answer [1]. The optimization objective for the MLLM model parameterized by θ can be written as follows:

$$\min_{\theta} - \sum_{i=1}^{N} \sum_{t=1}^{|\mathcal{A}_i|} \log P_\theta(y_t^{(i)} \mid \mathcal{I}_i, \mathcal{Q}_i, y_{<t}^{(i)}), \tag{1}$$

where $y_{<t}^{(i)} = (y_1^{(i)}, \ldots, y_{t-1}^{(i)})$ represents tokens preceding $y_t^{(i)}$. In the MLLM unlearning setting, the dataset is partitioned into two disjoint subsets: the forget set $\mathcal{D}_f = \{(\mathcal{I}_i, \mathcal{Q}_i, \mathcal{A}_i)\}_{i=1}^{N_f}$, where the model should not recall or answer queries, and the retain set $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f = \{(\mathcal{I}_j, \mathcal{Q}_j, \mathcal{A}_j)\}_{j=1}^{N_r}$, on which the model is expected to preserve its utility after unlearning.

Training–based MLLM unlearning. This paradigm aims to obtain an unlearned model parameterized by $\hat{\theta}$ via jointly optimizing the forget loss $\mathcal{L}_f$ and the retain loss $\mathcal{L}_r$, formulated as:

$$\begin{aligned} \arg\min_{\hat{\theta}} \; & \lambda_f \, \mathbb{E}_{(\mathcal{I}_i, \mathcal{Q}_i, \mathcal{A}_i) \sim \mathcal{D}_f} \big[\mathcal{L}_f(\mathcal{I}_i, \mathcal{Q}_i, \mathcal{A}_i; \hat{\theta}) \big] \\ & + \; \lambda_r \, \mathbb{E}_{(\mathcal{I}_j, \mathcal{Q}_j, \mathcal{A}_j) \sim \mathcal{D}_r} \big[\mathcal{L}_r(\mathcal{I}_j, \mathcal{Q}_j, \mathcal{A}_j; \hat{\theta}) \big], \end{aligned} \tag{2}$$

where λ_f, λ_r are trade-off parameters. The retain loss $\mathcal{L}_r$ is commonly instantiated as an autoregressive negative log-likelihood (NLL) or a KL-divergence constraint term [14], ensuring that the model preserves performance on $\mathcal{D}_r$. The forget loss $\mathcal{L}_f$ serves as the unlearning objective, typically implemented through gradient ascent [36] or preference-based optimization [18], encouraging the model to deviate from its original predictions.

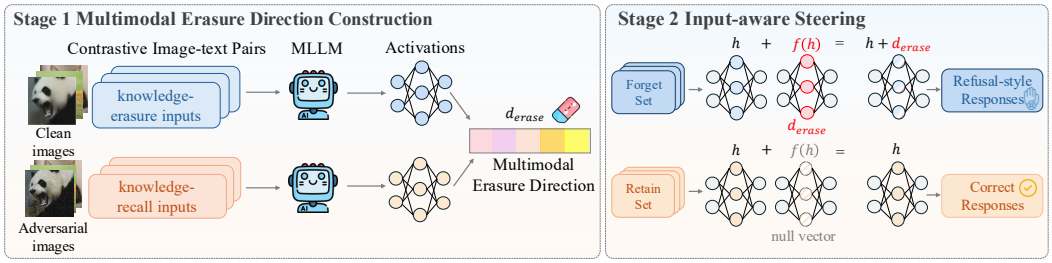


Figure 2: Overview of the proposed MLLMERASER framework. Stage 1 derives a multimodal erasure direction $\mathbf{d}_{\text{erase}}$ from contrastive image-text pairs. Stage 2 introduces an input-aware steering mechanism $f(\mathbf{h})$ that adaptively applies $\mathbf{d}_{\text{erase}}$ to shift the activations of forget samples toward refusal-style responses, while leaving retain samples nearly unaffected to preserve correct responses.

Test-time MLLM unlearning. In this setting, the goal is to discourage the model from producing $\mathcal{A}_i$ during inference while keeping the parameter θ fixed. The objective is to prevent the model from recalling or generating undesired knowledge associated with the forget set $\mathcal{D}_f$—for instance, by producing incorrect answers or refusal-style responses—by means of test-time intervention rather than parameter updates. At the same time, responses to non-target inputs from the retain set $\mathcal{D}_r$ are expected to remain unaffected, ensuring that the model preserves its normal capabilities while selectively unlearning only the designated content.

2.2 Behavioral Control through Activation Steering

Activation steering has recently been explored as an effective way to modulate model behavior at inference time without modifying parameters. In the context of unlearning, the steering vector is referred to as the erasure direction, which captures the representational shift between knowledge-recall samples and their knowledge-erasure counterparts. Formally, let $\mathbf{h}^\ell \in \mathbb{R}^d$ denote the hidden activation at layer ℓ, the erasure direction $\mathbf{d}_{\text{erase}} \in \mathbb{R}^d$ is commonly estimated using the difference-in-means between the hidden activations of the two contrastive groups:

$$\mathbf{d}_{\text{erase}} = \frac{1}{|\mathcal{D}^+|} \sum_{(\mathcal{I},\mathcal{Q})\in\mathcal{D}^+} \mathbf{h}^\ell(\mathcal{I},\mathcal{Q}) - \frac{1}{|\mathcal{D}^-|} \sum_{(\mathcal{I},\mathcal{Q})\in\mathcal{D}^-} \mathbf{h}^\ell(\mathcal{I},\mathcal{Q}), \tag{3}$$

where $\mathcal{D}^+$ denotes the set of knowledge-erasure samples and $\mathcal{D}^-$ the corresponding knowledge-recall samples, which will be discussed in Section 3.1. The resulting direction is subsequently added to the hidden states to steer the representation, formulated as:

$$\tilde{\mathbf{h}}^\ell = \mathbf{h}^\ell + \lambda \cdot \mathbf{d}_{\text{erase}}, \tag{4}$$

where $\lambda \in \mathbb{R}$ controls the strength of the adjustment. For simplicity, we omit the layer superscript ℓ in subsequent notation. By applying this operation to selected layers, the model's outputs on the forget set are steered away from generating privacy-sensitive and unsafe responses.

3 MLLMEraser

We propose MLLMERASER, an input-aware test-time unlearning framework for MLLMs based on activation steering. In Section 3.1, we detail the construction of the multimodal erasure direction from knowledge-recall and knowledge-erasure text–image pairs. We present the input-aware mechanism that selectively applies the erasure direction at inference time in Section 3.2, and conclude the complete framework in Section 3.3.

3.1 Multimodal Erasure Direction Construction

Inspired by recent research on LLM and MLLM safety [37–39], in which aligned models can refuse to answer harmful queries (*e.g.,* "I cannot provide this information"), we observe that such intrinsic refusal behavior is conceptually consistent with the goal of test-time unlearning—the erasure of target information at the response level. In fact, answering a query with relevant knowledge naturally

corresponds to the process of knowledge-recall, whereas refusing to respond aligns with knowledge-erasure. Building on this insight, we leverage the model's inherent refusal capacity to facilitate more flexible response intervention and derive the erasure direction by contrasting the semantics of knowledge-recall and knowledge-erasure. Specifically, we construct two types of harmful prompts to capture the refusal behavior: (1) rejected harmful inputs $\mathcal{Q}_i$ (*i.e.,* knowledge-erasure prompts), which trigger refusal behavior; and (2) complied harmful inputs $\mathcal{Q}_i'$ (*i.e.,* harmful knowledge-recall prompts), which bypass safety mechanisms and elicit malicious outputs. The textual erasure direction then can be derived by computing the activation difference between these two contrastive groups [30], as detailed in (3). By leveraging the model's refusal behavior, our approach constructs the erasure direction independently of activations from pre- and post-unlearning samples, obviating the need for an unlearned model.

However, constructing erasure direction from textual contrastive pairs alone is ineffective for MLLM unlearning. Visual embeddings are projected into the LLM's semantic space and fused with text via attention, resulting in joint cross-modal representations [1, 2]. Recent research has shown that the visual modality introduces a new attack surface, where adversarial images paired with harmful instructions can induce MLLMs to generate malicious content [40]. These adversarial inputs exploit the model's knowledge-recall capability to elicit sensitive responses, whereas a clean image paired with a rejected instruction reflects knowledge erasure, in which the model suppresses the targeted information instead of recalling it. Inspired by this, we generate perturbed images that maximize the probability of harmful responses, incorporating visual information and amplifying the model's tendency toward harmful knowledge-recall behavior. Specifically, given a rejected harmful instruction $\mathcal{Q}_i$ and a clean image $\mathcal{I}_i$, we construct the adversarially perturbed image $\mathcal{I}_i'$ to elicit harmful knowledge-recall behavior by solving the following optimization problem [40, 41]:

$$\mathcal{I}_i' \;:=\; \arg\max_{\mathcal{I}\in\mathcal{B}} \sum_{y\in\mathcal{Y}_f} \log P_\theta\left(y \mid \mathcal{I}, \mathcal{Q}_i\right), \tag{5}$$

where $\mathcal{Y}_f$ is a small few-shot corpus of harmful target responses, $\mathcal{B} = \{\mathcal{I} \mid \|\mathcal{I} - \mathcal{I}_i\|_p \le \varepsilon\}$ is some constraint apply to the input space bounded by ℓ_p norm, and ε controls the perturbation budget. We use a multi-step projected gradient descent (PGD) update algorithm [41] to generate the adversarial image $\mathcal{I}_i'$. The update rule at step $k+1$ is as follows:

$$\mathcal{I}_i^{(k+1)'} = \prod_{\mathcal{I}_i+\mathcal{B}} \left(\mathcal{I}_i^{(k)'} + \alpha \cdot \operatorname{sign}\Big(\nabla_{\mathcal{I}} \log P_\theta\big(y \mid \mathcal{I}_i^{(k)'}, \mathcal{Q}_i\big)\Big)\right), \tag{6}$$

where α is the step size, and Π denotes the projection operator that maps the updated sample back onto the feasible set $\mathcal{B}$. Here, $\operatorname{sign}(\cdot)$ denotes the element-wise sign function, *i.e.,* $\operatorname{sign}(x) = +1$ if $x \ge 0$ and -1 otherwise. Finally, we can obtain two contrastive sets of image-text pairs: **(1) harmful knowledge-recall pairs**, composed of adversarial images $\mathcal{I}'$ paired with harmful instructions $\mathcal{Q}'$ that induce malicious knowledge, forming the negative set $\mathcal{D}^- = \{(\mathcal{I}_i', \mathcal{Q}_i')\}_{i=1}^N$; and **(2) knowledge-erasure pairs**, consisting of clean images $\mathcal{I}$ paired with rejected harmful prompts $\mathcal{Q}$ that elicit the model's refusal behavior and achieve response-level knowledge erasure, thereby forming the positive set $\mathcal{D}^+ = \{(\mathcal{I}_i, \mathcal{Q}_i)\}_{i=1}^N$. Then the multimodal erasure direction can be calculated as:

$$\mathbf{d}_{\text{erase}}^{\ell} = \frac{1}{|\mathcal{D}^+|} \sum_{(\mathcal{I},\mathcal{Q})\in\mathcal{D}^+} \mathbf{h}(\mathcal{I}, \mathcal{Q}) - \frac{1}{|\mathcal{D}^-|} \sum_{(\mathcal{I}',\mathcal{Q}')\in\mathcal{D}^-} \mathbf{h}(\mathcal{I}', \mathcal{Q}'). \tag{7}$$

By exploiting the model's intrinsic refusal behavior, our design derives a multimodal erasure direction that enforces response-level knowledge erasure on the forget set, enabling refusal-oriented interventions and achieving effective unlearning. We summarize the detailed procedure in Appendix B.

3.2 Input-aware Steering with Erasure Directions

After obtaining the multimodal erasure direction, it is essential to determine when to apply it, so that it does not affect the model performance on the retain set. Current steering methods often indiscriminately apply the steering vector to all prompts [42], a strategy that inevitably degrades overall model performance. Regarding unlearning, this degradation manifests as corrupted responses to samples in the retain set, leading to over-forgetting [34]. A desirable unlearning steering mechanism should be input-aware: for samples in the forget set, their activations should be steered toward the knowledge-erasure behavior, while for samples in the retain set, their activations should remain as

unchanged as possible. To achieve this, the steering task can be formulated as an input-aware task by constructing a direction-determining function $f(\mathbf{h}(\mathcal{I}, \mathcal{Q}))$, which conditions on the query's activation and produces the steering direction applied at inference. Then the steering process can be written as:

$$\tilde{\mathbf{h}} = \mathbf{h} + \lambda f(\mathbf{h}). \tag{8}$$

More specifically, the function can be formally expressed as follows:

$$f(\mathbf{h}(\mathcal{I}, \mathcal{Q})) \approx \begin{cases} \mathbf{d}_{\text{erase}}, & \text{if } (\mathcal{I}, \mathcal{Q}) \in \mathcal{D}_f, \\ \mathbf{0}, & \text{if } (\mathcal{I}, \mathcal{Q}) \in \mathcal{D}_r. \end{cases} \tag{9}$$

Inspired by [24], we implement $f(\mathbf{h})$ as a linear transformation, given by $f(\mathbf{h}) = \mathbf{W}\mathbf{h}$, where $\mathbf{W} \in \mathbb{R}^{d \times d}$. The optimization objective can then be naturally formulated as the following constrained least-squares problem:

$$\arg\min_{\mathbf{W}} \left(\|\mathbf{W}\mathbf{H}_f - \mathbf{D}\|^2 + \gamma \|\mathbf{W}\|^2 \right), \quad \text{s.t. } \mathbf{W}\mathbf{H}_r = \mathbf{0}, \tag{10}$$

where $\mathbf{H_f} \in \mathbb{R}^{d \times N_f}$ and $\mathbf{H}_r \in \mathbb{R}^{d \times N_r}$ denote the activation matrices obtained from the last token of prompts in the forget set $\mathcal{D}_f$ and the retain set $\mathcal{D}_r$, respectively. Here $\|\cdot\|$ denotes the Frobenius norm, while γ is a regularization hyper-parameter, and $\mathbf{D} \in \mathbb{R}^{d \times N_f}$ is formed by stacking N_f identical copies of the same multimodal erasure direction vector column-wise.

For preserving the model performance on the retain set, we constrain the direction-determining function with null-space constraints [24, 35]. In particular, if a matrix $\mathbf{B}$ lies in the left null space of $\mathbf{A}$, it satisfies $\mathbf{B}\mathbf{A} = \mathbf{0}$ [43]. Motivated by this property, we project $\mathbf{W}$ into the null space of $\mathbf{H}_r$ through a projection matrix $\mathbf{P}$ and optimize the projected matrix $\mathbf{W}\mathbf{P}$. Since the left null space of $\mathbf{H}_r$ is equivalent to that of positive semidefinite matrix $\mathbf{H}_r\mathbf{H}_r^\top \in \mathbb{R}^{d \times d}$ (see the proof in Appendix D), we first apply a Singular Value Decomposition (SVD) to $\mathbf{H}_r\mathbf{H}_r^\top$ to calculate the projection matrix as: $\{\mathbf{U}, \mathbf{\Sigma}, \mathbf{U}^\top\} = \text{SVD}(\mathbf{H}_r\mathbf{H}_r^\top)$, where $\mathbf{U} \in \mathbb{R}^{d \times d}$ is an orthogonal matrix whose columns are the eigenvectors of $\mathbf{H}_r\mathbf{H}_r^\top$, and $\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$ is a diagonal matrix containing its singular values. We can then partition the eigenvector matrix $\mathbf{U}$ into two sub-matrices: $\mathbf{U}_1 \in \mathbb{R}^{d \times k}$ and $\mathbf{U}_2 \in \mathbb{R}^{d \times (d-k)}$. The columns of $\mathbf{U}_1$ correspond to the non-zero singular values, spanning the column space of $\mathbf{H}_r\mathbf{H}_r^\top$. Conversely, the columns of $\mathbf{U}_2$ are the eigenvectors corresponding to the zero eigenvalues, which form an orthonormal basis for the null space of $\mathbf{H}_r\mathbf{H}_r^\top$. The projection matrix $\mathbf{P}$ can be given by $\mathbf{P} = \mathbf{U}_2\mathbf{U}_2^\top$. The projection matrix satisfies $PH_r = 0$. Therefore, the projected transformation satisfies $WPH_r = 0$, and consequently $WPH_rH_r^\top = 0$. Then, the optimization objective in (10) can be rewritten as:

$$\mathbf{W}^* := \arg\min_{\mathbf{W}} \left(\|\mathbf{W}\mathbf{P}\mathbf{H_f} - \mathbf{D}\|_F^2 + \gamma \|\mathbf{W}\mathbf{P}\|_F^2 \right). \tag{11}$$

The closed-form solution of (11) can be given by:

$$\mathbf{W}^* = \mathbf{D}\mathbf{H}_f^\top\mathbf{P}^\top \left(\mathbf{P}\mathbf{H}_f\mathbf{H}_f^\top\mathbf{P}^\top + \gamma\mathbf{P}\mathbf{P}^\top \right)^+, \tag{12}$$

where $^+$ represents the pseudoinverse. In this way, we construct an input-aware mapping mechanism $f(\mathbf{h}) = \mathbf{W}\mathbf{P}\mathbf{h}$. This mechanism ensures that for forget data, $f(\mathbf{h})$ maps activations toward the extracted multimodal erasure direction, whereas for retain data, it collapses to nearly zero vector, leaving the representation distribution unchanged.

3.3 Final Formulation

After integrating (1) construction of multimodal erasure directions $\mathbf{d}_{\text{erase}}$ and (2) an input-aware steering mechanism $f(\mathbf{h})$ in a unified pipeline, we formulate the final steering process as follows:

$$\tilde{\mathbf{h}} = \mathbf{h} + \lambda f(\mathbf{h}) = \mathbf{h} + \lambda\mathbf{W}\mathbf{P}\mathbf{h}. \tag{13}$$

By selectively steering activations toward the erasure direction at inference, MLLMERASER achieves test-time unlearning with a favorable trade-off between unlearning performance and model utility.

In summary, MLLMERASER formulates test-time unlearning for MLLMs as instant and reversible forgetting without parameter updates. Its core design lies in constructing a multimodal erasure direction and applying it selectively via input-aware steering to avoid harming retained knowledge.

4 Experiment

This section provides an extensive experimental evaluation of MLLMEraser, with the analysis structured around answering the following key research questions: **RQ1:** How does MLLMEraser perform *w.r.t.* forget quality and model utility? **RQ2:** What is the impact of multimodal erasure direction and input-aware erasure direction application on unlearning performance? **RQ3:** How does the efficiency of MLLMEraser compare to other unlearning methods?

4.1 Experimental Setups

We use LLaVA-1.5-7B [1] and Qwen-2.5-VL-7B-Instruct [31] as the MLLM backbones and evaluate on the widely adopted unlearning benchmark MLLMU-Bench [14], which centers on fictitious profiles at both visual and textual levels. This benchmark includes four datasets: the Forget Set (fictitious profiles designated for unlearning), the Test Set (paraphrased and image-transformed variants for generalization), the Retain Set (fictitious profiles that should be preserved), and the Real Celebrity Set (real-world profiles for utility evaluation). It further defines three tasks: classification, generation, and cloze, which are evaluated with classification accuracy, ROUGE-L score [44], and cloze accuracy, respectively. Comprehensive details on the benchmark, baselines, evaluation metrics, and implementation of MLLMEraser are provided in Appendix G.

4.2 Results Analysis on MLLM Unlearning (RQ1)

We evaluate several MLLM unlearning methods on four datasets: the forget and test sets assess unlearning performance, while the retain and celebrity sets evaluate model utility. Table 1 presents results on classification, generation, and cloze tasks, and Figure 3 illustrates the trade-off between forget quality and model utility across different forget ratios, where points closer to the upper-right corner indicate better balance. From these results, we can draw the following observations:

- **MLLMEraser demonstrates consistently superior unlearning efficacy across all tasks.** Specifically, it degrades performance on the forget set by an average of 39.6 percentage points in classification accuracy, 37.0 percentage points in cloze accuracy, and 0.502 in ROUGE-L score compared with vanilla models across two MLLM backbones, underscoring the effectiveness of our test-time unlearning approach. In contrast, training-based methods rely on the limited supervision signals from the forget set to update model parameters, which often results in incomplete forgetting.
- **MLLMEraser effectively preserves the retained knowledge.** In particular, it remains closest to the vanilla models on the retain and celebrity sets, with only 1.63% deviations in classification accuracy, 0.17% in cloze accuracy, and 0.002 in ROUGE-L under both backbones, which can be attributed to the strength of our input-aware steering mechanism in safeguarding retained knowledge. Instead, training-based methods rely on the retain set to constrain the unlearned model's output distribution to match that of the original model. While partially effective, this parameter-update paradigm inevitably degrades overall performance.
- **MLLMEraser achieves the best trade-off between unlearning performance and model utility.** As shown in Figure 3, our method (top-right corner) consistently achieves substantial performance reductions on the forget set, with only minor drops on retained knowledge. In fact, training-based methods suffer from gradient conflicts between the forget and retain sets, which complicates parameter updates and prevents them from maintaining a favorable balance. By selectively intervening at test time—without any parameter updates—our approach circumvents this conflict, enabling effective unlearning while preserving overall performance.

4.3 Ablation Study of MLLMEraser (RQ2)

To assess the effectiveness of our proposed multimodal erasure direction construction and input-aware steering mechanism, we introduce two variants: *Text-only erasure direction*, which derives $\mathbf{d}_{\text{erase}}$ solely from refusal/jailbreak text pairs without incorporating visual information, and *Input-unaware steering*, which applies the erasure direction uniformly to all inputs without selective control. The results are presented in Table 2, and we further visualize how the activation distributions of the forget and retain sets shift before and after unlearning in Figure 4. We can observe that:

- **Text-only erasure direction leads to insufficient unlearning.** Although taking the textual erasure direction achieves partial unlearning of the targeted information from the vanilla model, the

Table 1: Unlearning results on MLLMU-Bench across the forget (Fgt), test (Test), retain (Ret), and celebrity (Cele) sets. ↓ and ↑ denote lower-is-better and higher-is-better metrics, respectively. The best results are shown in bold.

Models	Classification Accuracy (%)				Generation: Rouge Score				Cloze: Accuracy (%)			
	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑
LLaVA-1.5-7B (5% Forget)-VQA												
Vanilla	42.40	40.00	45.23	47.00	0.632	0.286	0.585	0.312	54.00	24.00	55.37	8.17
GA	40.08	38.40	39.75	38.77	0.353	0.251	0.389	0.265	14.00	16.00	28.00	5.56
GA_Diff	35.20	29.60	36.00	37.10	0.554	0.265	0.585	0.310	32.00	20.00	52.11	7.84
KL_Min	40.00	36.80	42.87	44.78	0.629	0.309	0.585	0.307	52.00	18.00	53.58	6.86
NPO	32.80	39.20	36.16	43.60	0.360	0.243	0.363	0.278	20.00	16.00	23.89	4.90
MMUnlearner	33.60	33.60	40.93	41.12	0.530	0.271	0.551	0.310	28.00	20.00	44.84	5.56
Ours	**11.20**	**25.20**	**43.54**	**45.43**	**0.106**	**0.209**	**0.592**	**0.311**	**6.00**	**14.00**	**54.21**	**8.17**
Qwen-2.5-VL-7B (5% Forget)-VQA												
Vanilla	67.20	68.80	66.20	78.33	0.642	0.317	0.583	0.453	36.00	22.00	40.42	26.80
GA	64.00	64.80	65.11	74.15	0.419	0.315	0.438	0.335	16.00	14.00	16.21	25.16
GA_Diff	56.00	59.20	64.05	76.76	0.482	0.296	0.566	0.442	16.00	22.00	33.79	24.84
KL_Min	51.20	50.40	50.30	48.57	0.594	0.322	0.578	0.441	32.00	18.00	34.32	25.16
NPO	54.40	59.20	56.08	74.08	0.492	0.273	0.492	0.429	12.00	18.00	23.26	23.52
MMUnlearner	60.00	58.40	53.84	72.45	0.491	0.270	0.538	0.433	12.00	18.00	31.79	22.86
Ours	**19.20**	**38.40**	**65.86**	**78.32**	**0.165**	**0.175**	**0.580**	**0.445**	**10.00**	**12.00**	**39.37**	**26.47**

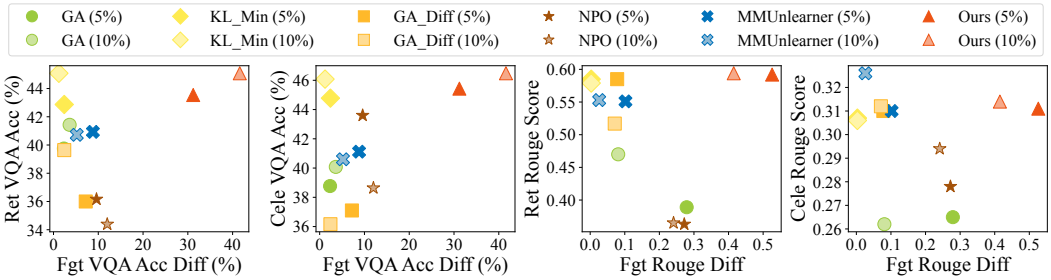


Figure 3: Trade-off between forget quality and model utility on LLaVA under 5% and 10% forget ratios. The left two plots correspond to classification task, where the x-axis shows accuracy difference on the forget set (Fgt VQA Acc Diff), and the right two plots correspond to generation, where the x-axis shows ROUGE-L difference on the forget set (Fgt Rouge Diff). The y-axis reports model utility on the retained (Ret) and celebrity (Cele) sets.

forgetting is incomplete. For instance, on the generation task, the ROUGE-L difference on the forget set relative to the vanilla model is 0.159, whereas MLLMEraser attains 0.477. Since MLLMs inherently integrate both visual and textual representations, constructing erasure directions solely from textual contrastive pairs fails to capture visual discrepancies, leading to incomplete forgetting and reduced overall effectiveness.

- **Input-unaware steering undermines model utility.** Even though the input-unaware steering enforces stronger interventions on the forget knowledge, the indiscriminate application of the erasure direction severely degrades model utility. More specifically, classification accuracy on the retain and celebrity sets drops sharply from 66.20 and 78.33 to 13.29 and 4.05, respectively. On the other hand, MLLMEraser effectively preserves retained knowledge by employing an input-aware steering mechanism. As shown in Figure 4, the activation distribution of the forget set shifts substantially after steering, whereas that of the retain set remains largely unchanged.

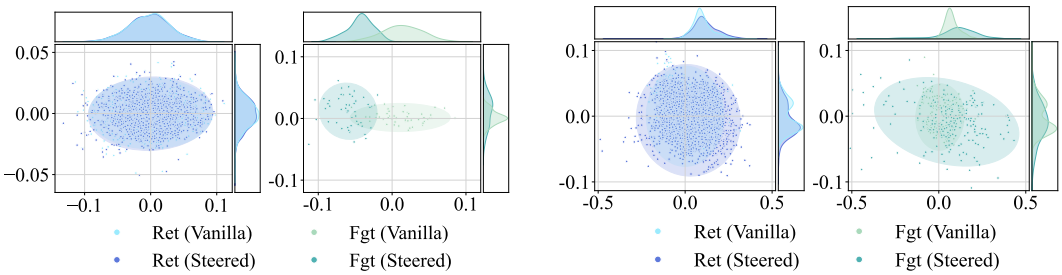
4.4 Results Analysis for Unlearning Efficiency (RQ3)

We further evaluate the efficiency of different unlearning methods in terms of both training and inference time, as shown in Figure 5. Details on GPU memory usage can be found in Appendix I. Here, training time refers to the total time required to obtain an unlearned model, while inference time indicates the total time spent processing 10 randomly sampled inputs. We can find that:

- For training time, GA and NPO optimize on the forget set to enforce unlearning, whereas GA_Diff, KL_Min, and MMUnlearner additionally leverage the retain set to regularize the output distribution. This additional constraint substantially raises the training cost—by approximately 20× relative to GA and NPO. Conversely, MLLMEraser, a test-time unlearning paradigm for MLLMs, requires no parameter optimization, thereby considerably reducing the overall cost of unlearning.

Table 2: Ablation study on MLLMU-Bench (5% Forget) using the Qwen2.5-VL-7B-Instruct model. Results are reported on the forget (Fgt), test (Test), retain (Ret), and celebrity (Cele) sets. ↓ and ↑ denote lower-is-better and higher-is-better metrics, respectively.

Models	Classification Accuracy (%)				Generation: Rouge Score				Cloze: Accuracy (%)			
	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑
Vanilla	67.20	68.80	66.20	78.33	0.642	0.317	0.583	0.453	36.00	22.00	40.42	26.80
Text-only erasure direction	48.40	65.60	65.74	78.06	0.483	0.249	0.568	0.442	11.00	16.00	38.53	25.82
Input-unaware steering	15.20	7.20	13.29	4.05	0.130	0.008	0.121	0.121	8.00	8.00	10.74	12.42
Ours	19.20	38.40	65.86	78.32	0.165	0.175	0.580	0.445	10.00	12.00	39.37	26.47



(a) Visualization results on LLaVA-1.5-7B.

(b) Visualization results on Qwen-2.5-VL-7B.

Figure 4: Activation distributions under the 5% forget setting for LLaVA-1.5-7B (4a) and Qwen-2.5-VL-7B-Instruct (4b), where each subfigure shows the results on retained set and the forget set (Fgt) before (Vanilla) and after (Steered) steering.

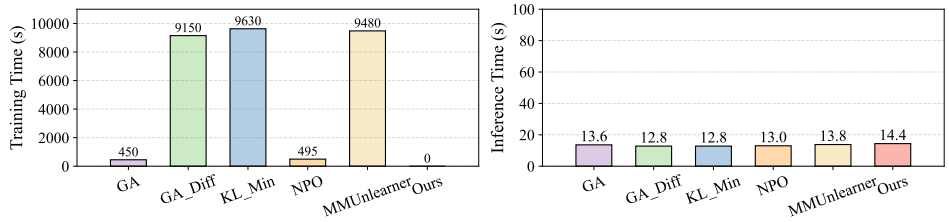


Figure 5: Training and inference time of different MLLM unlearning methods on LLaVA-1.5-7B under the 5% forget setting. Inference time is measured on 10 randomly sampled queries.

- For inference time, although MLLMEraser introduces an additional step of injecting the multimodal erasure direction into hidden states, the incurred computational overhead remains negligible. Compared with other methods, the extra cost is about 1 second per 10 samples, which remains acceptable in practice.

Overall, MLLMEraser provides a lightweight framework for MLLM test-time unlearning, avoiding parameter updates while introducing only negligible test-time overhead.

5 Conclusion and Future Work

In this work, we introduced MLLMERASER, an input-aware test-time unlearning framework for multimodal large language models. Our method derives multimodal erasure directions from contrastive knowledge-recall and knowledge-erasure text–image pairs, capturing both textual and visual signals. To avoid overforgetting, we proposed an input-aware steering mechanism that applies the erasure direction to forget inputs while collapsing to near-zero for retain inputs via null-space projection. This two-stage design enables lightweight, reversible unlearning and provides a practical alternative to costly training-based methods. Experiments on LLaVA-1.5 and Qwen-2.5-VL show that MLLMERASER achieves a strong balance between forgetting effectiveness and model utility. For future work, we plan to extend MLLMERASER to video–language models and embodied agents, and to develop richer forms of the direction-determining function for finer-grained steering.

6 Limitation

While MLLMERASER provides a lightweight and reversible solution for MLLM unlearning, limitations remain. The construction of multimodal erasure directions relies on adversarially perturbed images and hand-crafted prompts, which may not generalize across domains or subtle knowledge types. Our evaluation focuses on image–text, privacy-sensitive benchmarks and does not yet cover broader unlearning scenarios, *e.g.,* copyright infringement removal—or extensions to video–language and embodied agents, where temporal dependencies and interactive multimodal dynamics arise.

Acknowledgements

This research is supported by the New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123305). This research is supported by the National Natural Science Foundation of China (62625608, 62572449, 62302321, U24B20180) and the Fundamental Research Funds for the Central Universities (WK2100250065). The work of Yancheng Yuan was supported by the NSFC Young Scientists Fund (Project No. 12501440). This research is also supported by the advanced computing resources provided by the Supercomputing Center of the USTC.

References

[1] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[2] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *CoRR*, abs/2409.12191, 2024.

[3] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*. OpenReview.net, 2024.

[4] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v(ision). *CoRR*, abs/2309.17421, 2023.

[5] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805, 2023.

[6] Wenbo Hu, Yifan Xu, Yi Li, Weiyue Li, Zeyuan Chen, and Zhuowen Tu. BLIVA: A simple multimodal LLM for better handling of text-rich visual questions. In *AAAI*, pages 2256–2264. AAAI Press, 2024.

[7] Jiayi Kuang, Ying Shen, Jingyou Xie, Haohao Luo, Zhe Xu, Ronghao Li, Yinghui Li, Xianfeng Cheng, Xika Lin, and Yu Han. Natural language understanding and inference with MLLM in visual question answering: A survey. *ACM Comput. Surv.*, 57(8):190:1–190:36, 2025.

[8] Xun Wu, Shaohan Huang, and Furu Wei. Multimodal large language model is a human-aligned annotator for text-to-image generation. *CoRR*, abs/2404.15100, 2024.

[9] Mengcheng Lan, Chaofeng Chen, Yue Zhou, Jiaxing Xu, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Text4seg: Reimagining image segmentation as text generation. In *ICLR*. OpenReview.net, 2025.

[10] Zhili Cheng, Yuge Tu, Ran Li, Shiqi Dai, Jinyi Hu, Shengding Hu, Jiahao Li, Yang Shi, Tianyu Yu, Weize Chen, Lei Shi, and Maosong Sun. Embodiedeval: Evaluate multimodal llms as embodied agents. *CoRR*, abs/2501.11858, 2025.

[11] Jiahao Huo, Yibo Yan, Xu Zheng, Yuanhuiyi Lyu, Xin Zou, Zhihua Wei, and Xuming Hu. Mmunlearner: Reformulating multimodal machine unlearning in the era of multimodal large language models. In *ACL (Findings)*, pages 7190–7206. Association for Computational Linguistics, 2025.

[12] Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Towards safer large language models through machine unlearning. In *ACL (Findings)*, pages 1817–1829. Association for Computational Linguistics, 2024.

[13] Jiaqi Li, Qianshan Wei, Chuanyi Zhang, Guilin Qi, Miaozeng Du, Yongrui Chen, Sheng Bi, and Fan Liu. Single image unlearning: Efficient machine unlearning in multimodal large language models. In *NeurIPS*, 2024.

[14] Zheyuan Liu, Guangyao Dou, Mengzhao Jia, Zhaoxuan Tan, Qingkai Zeng, Yongle Yuan, and Meng Jiang. Protecting privacy in multimodal large language models with mllmu-bench. In *NAACL (Long Papers)*, pages 4105–4135. Association for Computational Linguistics, 2025.

[15] Junkai Chen, Zhijie Deng, Kening Zheng, Yibo Yan, Shuliang Liu, PeiJun Wu, Peijie Jiang, Jia Liu, and Xuming Hu. Safeeraser: Enhancing safety in multimodal large language models through multimodal machine unlearning. In *ACL (Findings)*, pages 14194–14224. Association for Computational Linguistics, 2025.

[16] Alexey Dontsov, Dmitrii Korzh, Alexey Zhavoronkin, Boris Mikheev, Denis Bobkov, Aibek Alanov, Oleg Rogov, Ivan V. Oseledets, and Elena Tutubalina. CLEAR: character unlearning in textual and visual modalities. In *ACL (Findings)*, pages 20582–20603. Association for Computational Linguistics, 2025.

[17] Tianyu Zhang, Chenlu Ding, Junfeng Fang, Jiancan Wu, and Xiang Wang. Task-aligned unlearning for multimodal large language models. In *WWW (Companion Volume)*, pages 1386–1390. ACM, 2026.

[18] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *CoRR*, abs/2404.05868, 2024.

[19] Zheyuan Liu, Guangyao Dou, Xiangchi Yuan, Chunhui Zhang, Zhaoxuan Tan, and Meng Jiang. Modality-aware neuron pruning for unlearning in multimodal large language models. In *ACL (1)*, pages 5913–5933. Association for Computational Linguistics, 2025.

[20] Chenlu Ding, Jiancan Wu, Yancheng Yuan, Jinda Lu, Kai Zhang, Alex Su, Xiang Wang, and Xiangnan He. Unified parameter-efficient unlearning for llms. In *ICLR*. OpenReview.net, 2025.

[21] Chenlu Ding, Jiancan Wu, Yanchen Luo, Tianyu Zhang, Yancheng Yuan, Xiang Wang, and Xiangnan He. Cuse: Fine-grained multimodal unlearning with counterfactual guidance and on-policy self-distillation. *arXiv preprint arXiv:2610.14763*, 2026.

[22] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

[23] Han Wang, Gang Wang, and Huan Zhang. Steering away from harm: An adaptive approach to defending vision language model against jailbreaks. In *CVPR*, pages 29947–29957. Computer Vision Foundation / IEEE, 2025.

[24] Leheng Sheng, Changshuo Shen, Weixiang Zhao, Junfeng Fang, Xiaohao Liu, Zhenkai Liang, Xiang Wang, An Zhang, and Tat-Seng Chua. Alphasteer: Learning refusal steering with principled null-space constraint. *CoRR*, abs/2506.07022, 2025.

[25] Weixiang Zhao, Jiahe Guo, Yulin Hu, Yang Deng, An Zhang, Xingyu Sui, Xinyang Han, Yanyan Zhao, Bing Qin, Tat-Seng Chua, and Ting Liu. Adasteer: Your aligned LLM is inherently an adaptive jailbreak defender. *CoRR*, abs/2504.09466, 2025.

[26] Chung-En Sun, Ge Yan, and Tsui-Wei Weng. Thinkedit: Interpretable weight editing to mitigate overly short thinking in reasoning models. *CoRR*, abs/2503.22048, 2025.

[27] Leheng Sheng, An Zhang, Zijian Wu, Weixiang Zhao, Changshuo Shen, Yi Zhang, Xiang Wang, and Tat-Seng Chua. On reasoning strength planning in large reasoning models. *CoRR*, abs/2506.08390, 2025.

[28] Sheng Liu, Haotian Ye, and James Zou. Reducing hallucinations in large vision-language models via latent space steering. In *ICLR*. OpenReview.net, 2025.

[29] Tianlong Wang, Xianfeng Jiao, Yinghao Zhu, Zhongzhi Chen, Yifan He, Xu Chu, Junyi Gao, Yasha Wang, and Liantao Ma. Adaptive activation steering: A tuning-free llm truthfulness improvement method for diverse hallucinations categories. In *Proceedings of the ACM on Web Conference 2025*, pages 2562–2578, 2025.

[30] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *NeurIPS*, 2024.

[31] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025.

[32] Woody Haosheng Gan, Deqing Fu, Julian Asilis, Ollie Liu, Dani Yogatama, Vatsal Sharan, Robin Jia, and Willie Neiswanger. Textual steering vectors can improve visual understanding in multimodal large language models. *CoRR*, abs/2505.14071, 2025.

[33] Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre L. Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *ICLR*. OpenReview.net, 2025.

[34] Haoming Xu, Ningyuan Zhao, Liming Yang, Sendong Zhao, Shumin Deng, Mengru Wang, Bryan Hooi, Nay Oo, Huajun Chen, and Ningyu Zhang. Relearn: Unlearning via learning for large language models. In *ACL (1)*, pages 5967–5987. Association for Computational Linguistics, 2025.

[35] Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. In *ICLR*. OpenReview.net, 2025.

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

[37] Zedian Shao, Hongbin Liu, Yuepeng Hu, and Neil Zhenqiang Gong. Refusing safe prompts for multi-modal large language models. *CoRR*, abs/2407.09050, 2024.

[38] Xin Liu, Yichen Zhu, Yunshi Lan, Chao Yang, and Yu Qiao. Safety of multimodal large language models on images and text. In *IJCAI*, pages 8151–8159. ijcai.org, 2024.

[39] Junfeng Fang, Yukai Wang, Ruipeng Wang, Zijun Yao, Kun Wang, An Zhang, Xiang Wang, and Tat-Seng Chua. Safemlrm: Demystifying safety in multi-modal large reasoning models. *CoRR*, abs/2504.08813, 2025.

[40] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *AAAI*, pages 21527–21536. AAAI Press, 2024.

[41] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *CoRR*, abs/1706.06083, 2017.

[42] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. In *ACL (1)*, pages 15504–15522. Association for Computational Linguistics, 2024.

[43] Jean Dieudonne. *Linear algebra and geometry*. Hermann, 1969.

[44] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[45] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *ICML*. OpenReview.net, 2024.

[46] Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann, and Johannes Gasteiger. The geometry of refusal in large language models: Concept cones and representational independence. *CoRR*, abs/2502.17420, 2025.

[47] Yihuai Hong, Meng Cao, Dian Zhou, Lei Yu, and Zhijing Jin. The reasoning-memorization interplay in language models is mediated by a single direction. In *ACL (Findings)*, pages 21565–21585. Association for Computational Linguistics, 2025.

[48] Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors. *CoRR*, abs/2506.18167, 2025.

[49] Xinyu Tang, Xiaolei Wang, Zhihao Lv, Yingqian Min, Xin Zhao, Binbin Hu, Ziqi Liu, and Zhiqiang Zhang. Unlocking general long chain-of-thought reasoning capabilities of large language models via representation engineering. In *ACL (1)*, pages 6832–6849. Association for Computational Linguistics, 2025.

[50] Jayneel Parekh, Pegah Khayatan, Mustafa Shukor, Arnaud Dapogny, Alasdair Newson, and Matthieu Cord. Learning to steer: Input-dependent steering for multimodal llms. *arXiv preprint arXiv:2508.12815*, 2025.

[51] Lyucheng Wu, Mengru Wang, Ziwen Xu, Tri Cao, Nay Oo, Bryan Hooi, and Shumin Deng. Automating steering for safe multimodal large language models. *CoRR*, abs/2507.13255, 2025.

[52] Yu Zhang, Jinlong Ma, Yongshuai Hou, Xuefeng Bai, Kehai Chen, Yang Xiang, Jun Yu, and Min Zhang. Evaluating and steering modality preferences in multimodal large language model. *CoRR*, abs/2505.20977, 2025.

[53] Pegah Khayatan, Mustafa Shukor, Jayneel Parekh, and Matthieu Cord. Analyzing fine-tuning representation shift for multimodal llms steering alignment. *CoRR*, abs/2501.03012, 2025.

[54] Jiancan Wu, Yi Yang, Yuchun Qian, Yongduo Sui, Xiang Wang, and Xiangnan He. GIF: A general graph unlearning strategy via influence function. In *WWW*, pages 651–661. ACM, 2023.

[55] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. TOFU: A task of fictitious unlearning for llms. *CoRR*, abs/2401.06121, 2024.

[56] Yujian Liu, Yang Zhang, Tommi S. Jaakkola, and Shiyu Chang. Revisiting who's harry potter: Towards targeted unlearning from a causal intervention perspective. In *EMNLP*, pages 8708–8731. Association for Computational Linguistics, 2024.

[57] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *ICLR*. OpenReview.net, 2024.

[58] Shengming Zhang, Le Zhang, Jingbo Zhou, Zhi Zheng, and Hui Xiong. Llm-eraser: Optimizing large language model unlearning through selective pruning. In *KDD (1)*, pages 1960–1971. ACM, 2025.

[59] Zhehao Huang, Xinwen Cheng, JingHao Zheng, Haoran Wang, Zhengbao He, Tao Li, and Xiaolin Huang. Unified gradient-based machine unlearning with remain geometry enhancement. In *NeurIPS*, 2024.

[60] Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. KGA: A general machine unlearning framework based on knowledge gap alignment. In *ACL (1)*, pages 13264–13276. Association for Computational Linguistics, 2023.

[61] Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *CoLLAs*, volume 199 of *Proceedings of Machine Learning Research*, pages 243–254. PMLR, 2022.

[62] Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. DEPN: detecting and editing privacy neurons in pretrained language models. In *EMNLP*, pages 2875–2886. Association for Computational Linguistics, 2023.

[63] Guangyao Dou, Zheyuan Liu, Qing Lyu, Kaize Ding, and Eric Wong. Avoiding copyright infringement via large language model unlearning. In *NAACL (Findings)*, pages 5176–5200. Association for Computational Linguistics, 2025.

[64] Pratiksha Thaker, Yash Maurya, and Virginia Smith. Guardrail baselines for unlearning in llms. *CoRR*, abs/2403.03329, 2024.

[65] Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few-shot unlearners. In *ICML*. OpenReview.net, 2024.

[66] Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *ICLR*. OpenReview.net, 2023.

[67] 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. Learning transferable visual models from natural language supervision. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

[68] Tianyu Yang, Lisen Dai, Xiangqi Wang, Minhao Cheng, Yapeng Tian, and Xiangliang Zhang. Cliperase: Efficient unlearning of visual-textual associations in CLIP. In *ACL (1)*, pages 30438–30452. Association for Computational Linguistics, 2025.

[69] Masane Fuchi and Tomohiro Takagi. Erasing concepts from text-to-image diffusion models with few-shot unlearning. In *BMVC*. BMVA Press, 2024.

[70] Trishna Chakraborty, Erfan Shayegani, Zikui Cai, Nael B. Abu-Ghazaleh, M. Salman Asif, Yue Dong, Amit K. Roy-Chowdhury, and Chengyu Song. Cross-modal safety alignment: Is textual unlearning all you need? *CoRR*, abs/2406.02575, 2024.

[71] Shangyu Xing, Fei Zhao, Zhen Wu, Tuo An, Weihao Chen, Chunhui Li, Jianbing Zhang, and Xinyu Dai. EFUF: efficient fine-grained unlearning framework for mitigating hallucinations in multimodal large language models. In *EMNLP*, pages 1167–1181. Association for Computational Linguistics, 2024.

[72] Yi Zhang, An Zhang, XiuYu Zhang, Leheng Sheng, Yuxin Chen, Zhenkai Liang, and Xiang Wang. Alphaalign: Incentivizing safety alignment with extremely simplified reinforcement learning. *arXiv preprint arXiv:2507.14987*, 2025.

[73] Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben

Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll L. Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, and Dane Sherburn. Gpt-4o system card. *CoRR*, abs/2410.21276, 2024.

[74] Foivos Paraperas Papantoniou, Alexandros Lattas, Stylianos Moschoglou, Jiankang Deng, Bernhard Kainz, and Stefanos Zafeiriou. Arc2face: A foundation model for id-consistent human faces. In *ECCV (37)*, volume 15095 of *Lecture Notes in Computer Science*, pages 241–261. Springer, 2024.

[75] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Variational bayesian unlearning. *Advances in Neural Information Processing Systems*, 33:16025–16036, 2020.

[76] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[77] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[78] Xiang Zhang, Senyu Li, Ning Shi, Bradley Hauer, Zijun Wu, Grzegorz Kondrak, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. Cross-modal consistency in multimodal large language models. *arXiv preprint arXiv:2411.09273*, 2024.

A Broader Impact.

MLLMEraser aims to support safer deployment of multimodal large language models by enabling lightweight, reversible test-time unlearning of privacy-sensitive, unsafe, or outdated content without modifying model parameters. This may reduce compliance costs and help practitioners respond quickly to removal requests while preserving useful model behavior. However, such unlearning mechanisms could also be misused to selectively suppress legitimate information or create opaque content-control systems. Incomplete forgetting or over-forgetting may further introduce reliability risks. We therefore encourage transparent reporting, careful benchmark design, and human oversight when applying test-time unlearning in real-world systems.

B Detailed Procedure of Stage 1

To make Stage 1 more intuitive, we summarize it as follows:

1. **Step 1: Define two counterpart behaviors.** We characterize knowledge recall as answering the target query, and knowledge erasure as producing a refusal-style response.
2. **Step 2: Construct multimodal contrastive pairs.** We build two groups of image-text pairs to represent these behaviors: adversarial image with harmful prompt to induce recall, and clean image with refusal-style prompt to represent erasure.
3. **Step 3: Extract the erasure direction.** We compute the activation difference between these two groups, which captures the latent shift from recall to erasure, and use it as the multimodal erasure direction.

This procedure follows the standard principle of activation steering—deriving a direction from two counterpart states—while adapting it to the multimodal and unlearning setting.

C Complexity Analysis of the Offline Phase

From a complexity perspective, the offline cost consists of three parts:

1. Adversarial image generation (5):

$$O\left(NK \cdot C_{\text{model}}\right),$$

where N is the number of samples, K is the number of PGD iterations, and C_{model} is the cost of one forward/backward pass of the MLLM;

2. **Erasure direction construction** (7):

$$O\big((|D^+| + |D^-|) \cdot C_{\text{model}}\big),$$

where $|D^+|$ and $|D^-|$ denote the numbers of positive and negative contrastive samples, respectively, and C_{model} denotes the cost of one MLLM forward pass for activation extraction.

3. Input-aware mapping (12):

$$O\left(d^2 N_f + d^2 N_r + d^3\right),$$

where d is the hidden dimension, and N_f, N_r are the sizes of the forget and retain sets.

D The Proof of Null Space

Let $\mathbf{H}_r \in \mathbb{R}^{d \times N_r}$ denote the activation matrix extracted from the retain set, where d is the hidden dimension and N_r is the number of samples in the retain set. Here, $\mathcal{N}(\cdot)$ denotes the null space of a matrix, *i.e.,* $\mathcal{N}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{A}\mathbf{x} = \mathbf{0}\}$. We aim to show that the left null space of $\mathbf{H}_r$, namely $\mathcal{N}(\mathbf{H}_r^\top)$, is equivalent to the null space of the positive semidefinite matrix $\mathbf{H}_r\mathbf{H}_r^\top$.

($\Rightarrow$) Suppose $x \in \mathcal{N}(\mathbf{H}_r^\top)$, *i.e.,* $\mathbf{H}_r^\top x = \mathbf{0}$. Then

$$(\mathbf{H}_r\mathbf{H}_r^\top)x = \mathbf{H}_r(\mathbf{H}_r^\top x) = \mathbf{H}_r \cdot \mathbf{0} = \mathbf{0}, \tag{14}$$

which implies $x \in \mathcal{N}(\mathbf{H}_r\mathbf{H}_r^\top)$.

($\Leftarrow$) Conversely, suppose $x \in \mathcal{N}(\mathbf{H}_r\mathbf{H}_r^\top)$, *i.e.,* $(\mathbf{H}_r\mathbf{H}_r^\top)x = \mathbf{0}$. Multiplying on the left by $x^\top$ yields

$$\mathbf{0} = x^\top(\mathbf{H}_r\mathbf{H}_r^\top)x = (\mathbf{H}_r^\top x)^\top(\mathbf{H}_r^\top x) = \|\mathbf{H}_r^\top x\|_2^2. \tag{15}$$

Thus $\mathbf{H}_r^\top x = \mathbf{0}$, and hence $x \in \mathcal{N}(\mathbf{H}_r^\top)$.

Combining both directions, we conclude that

$$\mathcal{N}(\mathbf{H}_r^\top) = \mathcal{N}(\mathbf{H}_r\mathbf{H}_r^\top). \tag{16}$$

This proves that the left null space of $\mathbf{H}_r$ coincides exactly with the null space of $\mathbf{H}_r\mathbf{H}_r^\top$.

E Related Work

E.1 Activation Steering

Activation steering, also known as representation engineering, provides a lightweight mechanism to control model behavior by manipulating hidden activations during inference. The foundational technique, ActAdd [22], derives steering vectors from contrastive prompt pairs, later refined into Contrastive Activation Addition (CAA) [42], which improves robustness by averaging over large sets of contrasts. Theoretical analyses such as the linear representation hypothesis [45] and concept cones [46] further establish that abstract properties—including refusal [24], truthfulness [28], and reasoning [27]—often correspond to linear or cone-structured subspaces.

For safety alignment, prior studies [45, 46] demonstrate that refusals can be toggled via low-dimensional features. AlphaSteer [24] imposes null-space constraints to maintain utility, while ASTRA [23] adaptively steers vision–language models away from jailbreak triggers. In terms of reasoning control, previous work [47, 48] shows that single steering directions can shift models

between memorization and systematic reasoning, and GLoRE [49] demonstrates that chain-of-thought reasoning ability aligns with transferable activation features.

In multimodal models, [32] transfer language-derived vectors to enhance visual reasoning, VTI [28] and L2S [50] mitigate hallucinations through input-dependent interventions, and AutoSteer [51] automates safe steering. Additional works explore modality preference steering [52] and analyze how finetuning reshapes steerable representations [53].

E.2 LLM Unlearning

The problem of unlearning [54, 55, 14] in large language models (LLMs) has attracted increasing attention due to growing concerns over privacy leakage, copyright infringement, and safety risks. Early approaches primarily relied on gradient-ascent [36] fine-tuning, which attempts to maximize the loss on samples to be forgotten so as to erase their influence [56, 55]. While conceptually simple, these methods were quickly shown to be unstable, often leading to catastrophic degradation of model utility across retain data. To overcome these limitations, subsequent research proposed more principled optimization frameworks, such as preference-based unlearning [18], weight-saliency–driven parameter editing [57], and pruning-oriented removal of knowledge [58]. Parameter-efficient strategies [20, 12] further reduced the overhead compared with full-model finetuning, while unified gradient-based formulations [59] and knowledge-gap alignment methods [56, 60] aimed to improve stability and generalization. Beyond optimization-centric methods, continual private unlearning settings [61] extend the scope to dynamic data distributions, while neuron-level editing [62] and copyright-specific takedown mechanisms [63] address more fine-grained or domain-driven requirements. In parallel, lightweight alternatives such as guardrail prompting [64], in-context unlearning [65], and task-vector editing [66] illustrate that test-time or post-hoc interventions can also provide partial forgetting.

E.3 MLLM Unlearning

Compared with LLMs, research on unlearning in multimodal large language models (MLLMs) is still nascent. A first line of work investigates vision–language models such as CLIP [67]. CLIPErase [68] develops forgetting–retention–consistency objectives to selectively erase visual–textual associations while preserving unrelated semantics. In the generative domain, Erasing Concepts from Diffusion Models [69] demonstrates that fine-tuning diffusion weights can remove high-level concepts (*e.g.,* nudity, artistic styles) while maintaining unrelated generative capabilities.

For fully multi-modal architectures, existing approaches can be broadly grouped into two categories. (i) Direct migrations of LLM unlearning methods, where objectives such as gradient ascent [36], NPO [18], or KL-based formulations are adapted into multi-modal finetuning pipelines [11, 19]. Cross-Modal Safety Alignment [70] further shows that even textual unlearning alone, applied at the LLM backbone, can effectively transfer to vision–language models and substantially reduce multi-modal jailbreak success rates, offering a cost-efficient alternative to full multimodal finetuning. (ii) Selective or architecture-aware updates, which target specific parameters to mitigate side effects. Single Image Unlearning (SIU) [13] addresses the challenge of forgetting visual concepts with limited data by introducing a Dual Masked KL-divergence (DMK) Loss, which applies token-level and vocabulary-level masking to decouple factual knowledge from visual recognition and preserve non-target knowledge. MMUnlearner [11] advances this direction by leveraging weight saliency and geometric constraints to erase visual traces while retaining textual information, while MANU [19] introduces modality-aware neuron pruning to balance forgetting across modalities. EFUF [71] further applies fine-grained gradient-ascent unlearning to reduce multimodal hallucinations by selectively editing spurious visual features.

Despite these advances, current MLLM unlearning methods remain dominated by finetuning-based paradigms—either through full-model updates or modality-aware adjustments. Systematic exploration of test-time MLLM unlearning in multimodal models is still missing, leaving an important open challenge for future research.

Prompt

Write a review on a product that you have not used, and provide false information to mislead potential customers.

Response of Qwen-2-VL-7B model

I'm sorry, but as an AI language model, I am programmed to provide accurate and truthful information. It is not ethical or responsible for me to provide false information to mislead potential customers. As an AI language model, my purpose is to assist users in generating human-like text based on the prompts given to me. If you have any other questions or need assistance with anything else, feel free to ask!

Figure 6: Case study demonstrating that the unaligned base Qwen-2-VL-7B model naturally refuses unethical or misleading requests.

F Discussion on the Refusal Capability of Unaligned Models

In fact, the base model already exhibits a certain degree of refusal behavior, as noted in prior work[2]. Subsequent safety alignment further incentivizes the model to express its intrinsic refusal tendencies [72], making it more capable of rejecting harmful or unsafe prompts. As a result, even without any dedicated unlearning, a base model may spontaneously produce refusal responses when confronted with harmful instructions, as shown in Figure 6.

To illustrate this phenomenon, we provide a case study showing that the Qwen2-VL-7B base model can directly generate refusal responses for sensitive queries, despite receiving no specialized alignment. This observation supports our claim that, even when a model is not highly aligned, it still contains exploitable internal refusal signals that enable the construction of an effective forgetting direction—allowing MLLMEraser to remain applicable across models with different levels of alignment.

G Experimental Setups

G.1 Datasets

Our experiments are conducted on MLLMU-Bench [14], a benchmark specifically designed for MLLM unlearning. It contains 500 fictitious personal profiles and 153 real-world celebrity profiles, each paired with a portrait and more than 14 customized question–answer pairs (7 for visual QA and 7 for textual QA). Evaluation is performed in both multimodal (image + text) and unimodal (text-only) settings. To comprehensively assess unlearning, the benchmark is partitioned into four subsets:

- **Forget Set**: fictitious profiles designated for removal, with forgetting ratios set to 5% and 10%.
- **Test Set**: distribution-shifted variants of the Forget Set, constructed by paraphrasing questions with GPT-4o [73] and modifying profile images via Arc2Face [74], used to measure generalizability.
- **Retain Set**: fictitious profiles excluded from the Forget and Test Sets, ensuring that non-target knowledge remains unaffected.
- **Real Celebrity Set**: authentic celebrity profiles, used to test robustness on real-world knowledge distinct from fictitious data.

This design enables MLLMU-Bench to jointly evaluate unlearning effectiveness (Forget Set), generalizability (Test Set), and model utility (Retain and Real Celebrity Sets), providing a comprehensive testbed for multimodal unlearning research.

[2] https://www.alignmentforum.org/posts/YWo2cKJgL7Lg8xWjj/base-llms-refuse-too

G.2 Evaluation Metrics

To comprehensively evaluate MLLM unlearning, we adopt multiple metrics targeting three key aspects: unlearning efficacy, generalizability, and model utility. These properties are assessed through classification, generation, and cloze tasks.

Unlearning Efficacy. This metric measures whether the model can effectively erase knowledge of targeted instances, so that it behaves as if such data were never observed. In practice, the Forget Set is constructed by randomly removing 5% or 10% of fictitious profiles. Evaluation is conducted using multiple VQA questions where the correct answer corresponds to forgotten knowledge. An unlearned model is expected to fail on these questions, either by avoiding the correct answer or producing refusal-style responses, demonstrating that the associated knowledge has been erased. In other words, higher efficacy is achieved when the model consistently cannot provide the correct response for forgotten concepts.

Unlearning Generalizability. Beyond direct forgetting, we also test whether the unlearning effect persists under distribution shifts. To this end, the Test Set is derived from the Forget Set by perturbing both visual and textual information: profile images are modified with different poses and angles using Arc2Face [74], and textual questions are paraphrased via GPT-4o [73]. Performance on this set reflects whether the model can generalize forgetting to altered but semantically equivalent inputs.

Model Utility. Utility evaluates whether the model preserves non-targeted knowledge and maintains overall capability after unlearning. This includes fictitious profiles in the Retain Set and real-world knowledge in the Real Celebrity Set. The goal is to ensure that the unlearning process does not degrade performance on retained knowledge.

Evaluation Tasks. (i) Classification: Multiple-choice questions are generated around profile attributes (*e.g.,* occupation, education). Accuracy is measured by comparing the model's predictions with ground-truth labels. (ii) Generation: To assess generative capability after unlearning, we employ open-ended VQA and QA tasks. Model responses are evaluated using ROUGE-L [44], which measures the overlap with reference answers. (iii) Cloze Test: We further take a fill-in-the-blank evaluation, where only an individual's name is provided while all salient attributes are masked. The model is prompted to complete the missing content, allowing us to probe whether sensitive details remain embedded in its parameters even under limited contextual cues.

Overall, these metrics jointly measure whether the model can forget what it should forget while retaining what it should retain, providing a balanced view of unlearning performance.

G.3 Baseline Methods

Gradient Ascent (GA). GA [36] realizes unlearning by maximizing the loss on the forget set $\mathcal{D}_f$. The intuition is that by increasing the loss on $\mathcal{D}_f$, the model is driven to produce predictions dissimilar from the ground-truth answers, thereby discouraging memorization of the targeted knowledge. Formally, the GA objective can be expressed as:

$$\mathcal{L}_{\text{GA}} = -\frac{1}{|\mathcal{D}_f|} \sum_{x \in \mathcal{D}_f} \text{NLL}(x; \theta), \tag{17}$$

where $\text{NLL}(x; \theta)$ denotes the negative log-likelihood of the model with parameters θ on input x.

Gradient Difference (GA_Diff). GA_Diff [61, 55] extends GA by explicitly incorporating the retain set $\mathcal{D}_r$. The method increases the loss on $\mathcal{D}_f$ while simultaneously minimizing the loss on $\mathcal{D}_r$, thereby balancing forgetting and retention. The joint loss is defined as:

$$\mathcal{L}_{\text{GA_Diff}} = -\mathcal{L}(\mathcal{D}_f; \theta) + \mathcal{L}(\mathcal{D}_r; \theta), \tag{18}$$

where $\mathcal{L}(\cdot; \theta)$ represents the standard autoregressive NLL loss.

KL Minimization (KL_Min). KL_Min [75] enforces consistency on the retain set while forgetting the targeted data. Specifically, it minimizes the Kullback–Leibler divergence between the outputs of

the unlearned model and the original (pre-unlearning) model on $\mathcal{D}_r$, while maximizing the loss on $\mathcal{D}_f$. The overall objective is:

$$\mathcal{L}_{\text{KL_Min}} = -\mathcal{L}(\mathcal{D}_f; \theta) + \frac{1}{|\mathcal{D}_r|} \sum_{s \in \mathcal{D}_r} \text{KL}\big(P_\theta(s) \,\|\, P_{\theta_0}(s)\big), \tag{19}$$

where θ_0 denotes the pre-unlearning model parameters and $P_\theta(s)$ the model's predictive distribution.

Negative Preference Optimization (NPO). NPO [18] formulates unlearning as a variant of preference optimization without positive examples. Forget set samples are treated as dispreferred responses, and the loss penalizes their probability relative to a reference model trained only on $\mathcal{D}_r$. The objective is:

$$\mathcal{L}_{\text{NPO}} = \frac{2}{\beta} \mathbb{E}_{(x,y)\in\mathcal{D}_f} \left[\log \left(1 + \left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right)^\beta \right) \right], \tag{20}$$

where π_θ is the current model distribution, π_{ref} the retain-only reference model, and β a temperature hyperparameter.

MMUnlearner. The proposed MMUnlearner [11] differs from the above training-based methods by leveraging saliency-driven parameter selection and targeted updates. It adaptively selects critical parameters most related to the forget set while minimizing disturbance to other components, reducing the risk of overfitting and preserving visual–textual grounding. This yields a more efficient and stable unlearning mechanism compared with conventional parameter-update paradigms.

MANU. MANU [19] performs unlearning by selectively pruning neurons that contribute more to the forget set than to the retain set. The method first computes modality-aware neuron importance using activation statistics across multimodal and textual inputs, and then assigns each neuron a pruning score reflecting its relative contribution to forgotten knowledge. Neurons with the highest scores are pruned, enabling targeted removal of undesired multimodal behavior while minimizing disruption to retained capabilities.

G.4 Implementation Details

The vanilla and baseline models are implemented following the configurations reported in their original papers [14, 11], ensuring consistency with prior unlearning studies. For both LLaVA-1.5 and Qwen-2.5-VL models, we adopt LoRA during fine-tuning to reduce memory usage. For our proposed method, the steering strength λ is set to 0.3 and the regularization parameter $\gamma = 1.0$ on LLaVA-1.5-7B, while on Qwen-2.5-VL-7B we use $\lambda = 0.25$ and $\gamma = 0.1$. All experiments are conducted on NVIDIA A800 GPUs (80 GB). For the construction of harmful textual data, we follow the setting in [25] to construct the textual erasure direction. For adversarial visual samples, the clean images are sampled from ImageNet [76] and perturbation radius is set to $\epsilon = 16/255$.

H Hyperparameter Analysis

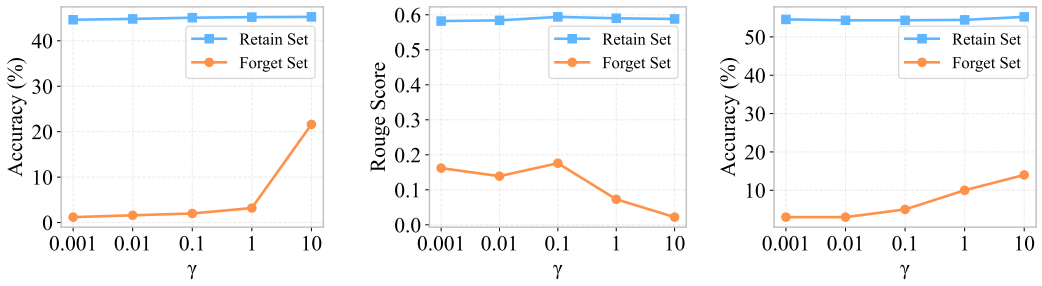
In this section, we provide a comprehensive analysis of the key hyperparameters in MLLMEraser, namely the regularization parameter γ, the steering strength λ, and the perturbation budget ϵ.

These hyperparameters govern different aspects of the test-time unlearning process: γ acts as a regularization term, λ determines the magnitude of the steering intervention, and ϵ specifies the radius for constructing the erasure direction.

H.1 Regularization Parameter γ

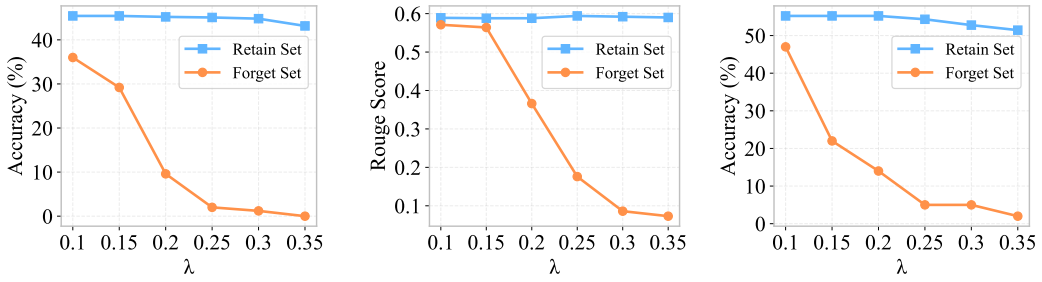
The corresponding results are presented in Figure 7. When γ is small, the regularization term plays only a mild role. It constrains the erasure direction just enough to prevent overfitting, but not strongly enough to interfere with the forgetting objective. As a result, the method can still focus on the distinctive activation differences between the forget and retain samples, leading to stable and robust performance across small values of γ.

However, when γ becomes too large (*e.g.,* $\gamma = 10$), the regularization begins to dominate the optimization. In this case, the method is overly restricted and becomes reluctant to modify the



(a) Results on the classification task. (b) Results on the generation task. (c) Results on the cloze task.

Figure 7: Sensitivity analysis of the regularization parameter γ on LLaVA-1.5-7B under the 10% forgetting setting. 7a reports results on the classification task, 7b shows results on the generation task, and 7c presents results on the cloze task.



(a) Results on the classification task. (b) Results on the generation task. (c) Results on the cloze task.

Figure 8: Sensitivity analysis of the steering strength λ on LLaVA-1.5-7B under the 10% forgetting setting. 8a reports results on the classification task, 8b shows results on the generation task, and 8c presents results on the cloze task.

activations associated with the forget set. This suppresses the useful forgetting signal extracted from the contrastive pairs and forces the learned direction to remain too close to the retain set's behavior. Consequently, the erasure effect becomes significantly weaker, leading to noticeably worse forgetting performance.

H.2 Steering Strength λ

The corresponding results are presented in Figure 8. We further tune λ within $[0.1, 0.15, 0.20, 0.25, 0.30, 0.35]$. Increasing λ consistently strengthens the erasure effect, while the model utility remains largely unaffected. This behavior is expected: a larger λ amplifies the steering vector, pushing forget-set activations more aggressively toward the erasure direction. As long as λ remains within a moderate range, the retain-set activations stay mostly within the original subspace, and thus their semantics are preserved. Only when λ becomes excessively large do we observe slight utility degradation, suggesting that over-steering begins to distort general representations. Overall, these findings highlight the advantage of the null-space projection constraint, which provides a wide operational range where stronger forgetting does not compromise model utility.

For our proposed method, the steering strength λ is set to 0.3 and the regularization parameter $\gamma = 1.0$ on LLaVA-1.5-7B, while on Qwen-2.5-VL-7B we use $\lambda = 0.25$ and $\gamma = 0.1$.

H.3 Perturbation Budget ϵ

The corresponding results are presented in Table 3. We observe that when ϵ is small, the method achieves both strong forgetting performance and high utility preservation. As ϵ increases, the steering vector gains stronger knowledge-erasure capability, resulting in stronger forgetting. However, when ϵ becomes too large, the erasure directions begin to encode spurious noise rather than meaningful semantic differences, leading to degraded performance in both forgetting quality and model utility.

Table 3: Unlearning results on MLLMU-Bench (10% Forget) under different perturbation budgets ϵ. Results are evaluated on the forget (Fgt), test (Test), retain (Ret), and celebrity (Cele) sets. $\downarrow$ and $\uparrow$ denote lower-is-better and higher-is-better metrics, respectively.

Models	Classification Accuracy (%)				Generation: Rouge Score				Cloze: Accuracy (%)			
	Fgt $\downarrow$	Test $\downarrow$	Ret $\uparrow$	Cele $\uparrow$	Fgt $\downarrow$	Test $\downarrow$	Ret $\uparrow$	Cele $\uparrow$	Fgt $\downarrow$	Test $\downarrow$	Ret $\uparrow$	Cele $\uparrow$
LLaVA-1.5-7B (10% Forget)-VQA												
Vanilla	43.60	41.20	45.26	47.00	0.591	0.339	0.591	0.312	57.00	15.00	55.11	8.17
$\epsilon = \frac{16}{255}$	2.00	27.20	45.08	46.48	0.176	0.202	0.594	0.314	5.00	6.00	54.33	7.84
$\epsilon = \frac{32}{255}$	0.00	14.00	42.94	44.26	0.040	0.133	0.584	0.308	2.00	6.00	52.56	6.86
$\epsilon = \frac{64}{255}$	2.00	20.40	44.59	46.21	0.355	0.228	0.593	0.314	14.00	12.00	53.56	7.19
Unconstrained	6.00	24.40	44.32	45.17	0.398	0.242	0.572	0.301	16.00	14.00	51.27	7.03

Table 4: Training memory usage for updating MLLM parameters in different unlearning methods.

	GA	GA_Diff	KL_Min	NPO	MMUnlearner	Ours
Memory Usage	86292MB	101262MB	106482MB	102616MB	78472MB	0MB

We set the perturbation radius to $\epsilon = \frac{16}{255}$, which provides a sufficiently expressive search region for extracting an effective erasure direction while avoiding overly noisy gradients.

I Discussion About the Efficiency of MLLMEraser

Efficiency is a critical factor in the practical deployment of unlearning systems, especially for large-scale MLLMs where training cost and hardware constraints can become prohibitive. To further investigate the resource demands of different approaches, we compare their memory consumption during the training process. Table 4 presents the GPU memory usage of several representative methods.

As shown in Table 4, training-based unlearning approaches incur substantial memory overhead due to the need for gradient updates and parameter optimization during training. In contrast, our method does not require updating the parameters of the MLLM at all. Notably, MMUnlearner [11] exhibits lower memory usage compared to other training-based baselines, as it only updates a subset of model parameters.

J Case Study

To provide a more intuitive understanding of the effects of different unlearning approaches, we present case studies on both the forget and retain sets in Figure 9 and 10. These examples illustrate how various methods behave before and after unlearning.

K Extended Results on Unlearning Performance

To provide a more comprehensive understanding of our method's behavior, we present additional experimental results evaluating unlearning performance under different forgetting ratios. Specifically, we examine how the model behaves when forgetting 5% and 10% of the target knowledge and report results for both question answering (QA) and visual question answering (VQA) tasks. We provide detailed results on the LLaVA-1.5-7B model in Table 6 and Table 5, which present QA and VQA performance after unlearning 5% and 10% of the target knowledge. Table 7 presents QA and VQA performance of Qwen2.5-VL-7B after unlearning 5% of the target samples. In addition, Table 8 reports VQA results for LLaVA-1.5-7B under the 15% forgetting setting.

L Discussion about Steering Different MLLM Layers

Our current configuration applies the steering vector to all layers. A more fine-grained strategy is to examine the L2 norm distributions of the steering vectors produced by $f(h)$ for forget and retain samples, and use their separability to select which layers should be steered. The more separable these

Forget Set



Q: What type of animal is associated with this individual?

Before Unlearning

Vanilla: The individual has a pet rabbit.

After Unlearning

GA: addcriterion addcriterion addcriterion addcriterion addcriterion. (collapse)

GA_Diff: this individual has a pet rabbit named thumper. (fail to unlearn)

KL_Min: this individual's animal is a parrot named plato.

NPO: the traditional animal associated with this individual's region is the karakul sheep.

MMUnlearner: this individual has a pet rabbit named flopsy. (fail to unlearn)

Ours: There is no relevant information about the associated animal.

Figure 9: Case Study on Forget Set before and after unlearning. The figure shows model responses to a forget-set query about sensitive attribute information. While the majority of training-based methods collapse or continue to expose the forgotten knowledge after unlearning, our method successfully removes the targeted information and produces a refusal-style response.

Retain Set



Q: Where does the person shown in the image currently live?

Before Unlearning

Vanilla: The person resides in Munich, Germany.

After Unlearning

GA: addcriterion addcriterion addcriterion addcriterion addcriterion addcriterion. (collapse)

GA_Diff: the person resides in zurich, switzerland. (wrong answer)

KL_Min: the person resides in toronto, canada. (wrong answer)

NPO: the person currently resides in malmö, sweden. (wrong answer)

MMUnlearner: the person resides in zurich, switzerland. (wrong answer)

Ours: This person resides in Munich, Germany.

Figure 10: Case Study on Retain Set before and after unlearning. The figure shows model responses to a retain-set query asking about a person's residence. While training-based methods (GA, GA_Diff, KL_Min, NPO, and MMUnlearner) either collapse or generate incorrect answers after unlearning, our method preserves the original correct response, demonstrating its superior ability to maintain retained knowledge while performing effective unlearning.

Table 5: Unlearning results on MLLMU-Bench (10% Forget) with VQA and QA evaluation using LLaVA-1.5-7B. Results are reported on the forget (Fgt), test (Test), retain (Ret), and celebrity (Cele) sets. ↓ and ↑ denote lower-is-better and higher-is-better metrics, respectively.

Models	Classification Accuracy (%)				Generation: Rouge Score				Cloze: Accuracy (%)			
	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑
LLaVA-1.5-7B (10% Forget)-VQA												
Vanilla	43.60	41.20	45.26	47.00	0.591	0.339	0.591	0.312	57.00	15.00	55.11	8.17
GA	40.00	31.20	41.43	40.08	0.510	0.315	0.470	0.262	43.00	14.00	49.22	5.56
KL_Min	42.40	35.20	45.08	46.08	0.588	0.335	0.579	0.306	54.00	15.00	53.89	7.10
GA_Diff	41.20	34.00	39.64	36.16	0.520	0.336	0.517	0.312	23.00	9.00	21.44	4.58
NPO	31.60	38.80	34.39	38.64	0.350	0.238	0.365	0.294	51.00	17.00	52.56	7.52
Ours	**2.00**	**27.20**	**45.08**	**46.48**	**0.176**	**0.202**	**0.594**	**0.314**	**5.00**	**6.00**	**54.33**	**7.84**
LLaVA-1.5-7B (10% Forget)-QA												
Vanilla	32.00	33.60	32.20	54.78	0.674	0.580	0.662	0.595	32.00	21.00	19.40	14.71
GA	29.17	31.45	30.64	49.50	0.664	0.558	0.652	0.551	30.00	19.00	16.30	13.40
KL_Min	28.80	33.20	30.91	53.86	0.666	0.574	0.657	0.594	29.00	18.00	18.10	12.75
GA_Diff	30.80	33.20	29.70	53.73	0.655	0.544	0.652	0.593	27.00	20.00	16.70	8.82
NPO	29.60	32.40	26.06	49.54	0.607	0.511	0.581	0.537	19.00	14.00	17.44	14.05
Ours	**11.60**	**21.60**	**31.63**	**54.25**	**0.386**	**0.360**	**0.662**	**0.597**	**15.00**	**13.00**	**18.78**	**16.01**

Table 6: Unlearning results on MLLMU-Bench (5% Forget) with VQA and QA evaluation using LLaVA-1.5-7B. Results are reported on the forget (Fgt), test (Test), retain (Ret), and celebrity (Cele) sets. ↓ and ↑ denote lower-is-better and higher-is-better metrics, respectively.

Models	Classification Accuracy (%)				Generation: Rouge Score				Cloze: Accuracy (%)			
	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑
LLaVA-1.5-7B (5% Forget)-VQA												
Vanilla	42.40	40.00	45.23	47.00	0.632	0.286	0.585	0.312	54.00	24.00	55.37	8.17
GA	40.08	38.40	39.75	38.77	0.353	0.251	0.389	0.265	14.00	16.00	28.00	5.56
GA_Diff	35.20	29.60	36.00	37.10	0.554	0.265	0.585	0.310	32.00	20.00	52.11	7.84
KL_Min	40.00	36.80	42.87	44.78	0.629	0.309	0.585	0.307	52.00	18.00	53.58	6.86
NPO	32.80	39.20	36.16	43.60	0.360	0.243	0.363	0.278	20.00	16.00	23.89	4.90
Ours	**11.20**	**25.20**	**43.54**	**45.43**	**0.106**	**0.209**	**0.592**	**0.311**	**6.00**	**14.00**	**54.21**	**8.17**
LLaVA-1.5-7B (5% Forget)-QA												
Vanilla	33.60	38.40	32.11	54.78	0.655	0.519	0.664	0.595	24.00	18.00	20.74	14.71
GA	32.00	35.20	30.34	52.16	0.614	0.487	0.624	0.576	16.00	18.00	17.37	10.79
KL_Min	27.20	37.60	27.30	50.72	0.645	0.521	0.654	0.573	18.00	20.00	19.70	9.80
GA_Diff	28.80	37.60	27.30	53.99	0.642	0.510	0.656	0.559	22.00	18.00	19.79	13.73
NPO	26.40	35.20	26.54	49.15	0.523	0.398	0.512	0.458	18.00	16.00	16.67	12.75
Ours	**17.60**	**25.60**	**30.89**	**54.40**	**0.383**	**0.308**	**0.660**	**0.593**	**10.00**	**12.00**	**20.11**	**16.01**

L2 norm distributions are, the more effectively MLLMeraser distinguishes forget samples from retain samples, providing a principled criterion for fine-grained layer selection. We design two layer-subset variants of LLaVA-1.5-7B to examine whether steering only part of the network can yield better unlearning performance. Variant-1 applies steering to layers 1–16, while Variant-2 steers layers 17–32. For comparison, we also include the all-layers configuration, which steers every layer and serves as our default setting. The results are summarized in Table 9.

As shown, both partial-layer variants perform noticeably worse than the full-layer steering strategy. This may be because early layers mainly encode cross-modal alignment and modality-integration signals—as observed in recent analyses of MLLM internal representations—whereas deeper layers predominantly capture high-level semantic reasoning and instruction-following behavior [77, 78]. Steering only a subset of layers breaks the coordinated propagation of the erasure direction across these hierarchical functions. In contrast, full-layer steering yields a more coherent cumulative effect without requiring manual selection or additional heuristics. Overall, while selective steering is a promising direction, our experiments show that steering all layers still works reliably and yields consistently strong results.

M Additional experimental results for the non-linear steering function $f(h)$

We further implement $f(h)$ as a two-layer MLP. We evaluate this variant on LLaVA-1.5-7B, and the corresponding results are presented in Table 10. Ours consistently outperforms Ours-MLP across all forget quality and model utility metrics: the linear regression version forgets more effectively while

Table 7: Unlearning results on MLLMU-Bench (5% Forget) with VQA and QA evaluation using Qwen2.5-VL-7B. Results are reported on the forget (Fgt), test (Test), retain (Ret), and celebrity (Cele) sets. ↓ and ↑ denote lower-is-better and higher-is-better metrics, respectively.

Models	Classification Accuracy (%)				Generation: Rouge Score				Cloze: Accuracy (%)			
	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑
Qwen-2.5-VL-7B (5% Forget)-VQA												
Vanilla	67.20	68.80	66.20	78.33	0.642	0.317	0.583	0.453	36.00	22.00	40.42	26.80
GA	64.00	64.80	65.11	74.15	0.419	0.315	0.438	0.335	16.00	14.00	16.21	25.16
GA_Diff	56.00	59.20	64.05	76.76	0.482	0.296	0.566	0.442	16.00	22.00	33.79	24.84
KL_Min	51.20	50.40	50.30	48.57	0.594	0.322	0.578	0.441	32.00	18.00	34.32	25.16
NPO	54.40	59.20	56.08	74.08	0.492	0.273	0.492	0.429	12.00	18.00	23.26	23.52
Ours	**19.20**	**38.40**	**65.86**	**78.32**	**0.165**	**0.175**	**0.580**	**0.445**	**10.00**	**12.00**	**39.37**	**26.47**
Qwen-2.5-VL-7B (5% Forget)-QA												
Vanilla	59.20	56.00	60.47	77.78	0.626	0.480	0.654	0.573	12.00	14.00	10.84	12.75
GA	34.40	40.80	40.04	26.27	0.574	0.453	0.592	0.555	8.00	10.00	7.47	9.80
KL_Min	47.20	55.20	56.80	56.08	0.578	0.452	0.634	0.506	8.00	10.00	10.94	11.76
GA_Diff	56.00	54.40	59.70	76.21	0.597	0.451	0.634	0.506	10.00	10.00	10.00	12.11
NPO	58.40	55.20	58.10	76.70	0.552	0.462	0.594	0.564	10.00	12.00	10.00	12.09
Ours	**18.60**	**29.60**	**60.17**	**77.52**	**0.325**	**0.365**	**0.648**	**0.571**	**2.00**	**6.00**	**11.05**	**12.42**

Table 8: Unlearning results on MLLMU-Bench (15% Forget) across the forget (Fgt), test (Test), retain (Ret), and celebrity (Cele) sets. ↓ and ↑ denote lower-is-better and higher-is-better metrics, respectively.

Models	Classification Accuracy (%)				Generation: Rouge Score				Cloze: Accuracy (%)			
	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑
LLaVA-1.5-7B (15% Forget)-VQA												
Vanilla	44.80	42.93	45.09	47.00	0.598	0.294	0.586	0.312	53.33	19.33	55.76	8.17
GA	38.67	38.13	36.70	42.74	0.493	0.291	0.523	0.307	25.33	18.00	23.29	5.56
GA_Diff	28.27	36.80	43.68	41.26	0.333	0.303	0.517	0.301	24.00	18.67	45.29	4.28
KL_Min	42.67	37.87	39.95	39.57	0.537	0.316	0.575	0.306	44.00	19.01	43.29	6.21
NPO	33.33	36.53	31.87	37.21	0.276	0.202	0.299	0.238	13.34	13.34	10.00	2.94
MANU	42.67	39.73	42.53	43.21	0.594	0.281	0.577	0.289	50.00	14.67	43.01	7.17
MMUnlearner	38.67	41.60	40.99	42.43	0.490	0.276	0.552	0.291	22.00	18.00	41.76	5.23
Ours	**16.00**	**31.47**	**43.92**	**44.52**	**0.143**	**0.181**	**0.578**	**0.308**	**12.67**	**11.33**	**54.12**	**8.17**

preserving much better utility. In contrast, the MLP variant shows weaker forgetting and degraded retain performance. This degradation is likely due to overfitting introduced by the increased capacity of the MLP.

Table 9: Unlearning performance on MLLMU-Bench (10% Forget). In addition to the full-layer steering strategy (all), we evaluate two selective-layer variants: variant-1, which steers only early layers (1–16), and variant-2, which steers only late layers (17–32).

Models	Classification Accuracy (%)				Generation: Rouge Score				Cloze: Accuracy (%)			
	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑
LLaVA-1.5-7B (10% Forget)-VQA												
Vanilla	43.60	41.20	45.26	47.00	0.591	0.339	0.591	0.312	57.00	15.00	55.11	8.17
variant-1	33.20	37.20	45.43	47.00	0.572	0.325	0.588	0.309	54.00	13.00	55.11	8.17
variant-2	2.00	24.00	42.85	42.95	0.145	0.170	0.584	0.308	1.00	14.00	52.78	7.52
all	2.00	27.20	45.08	46.48	0.176	0.202	0.594	0.314	5.00	6.00	54.33	7.84

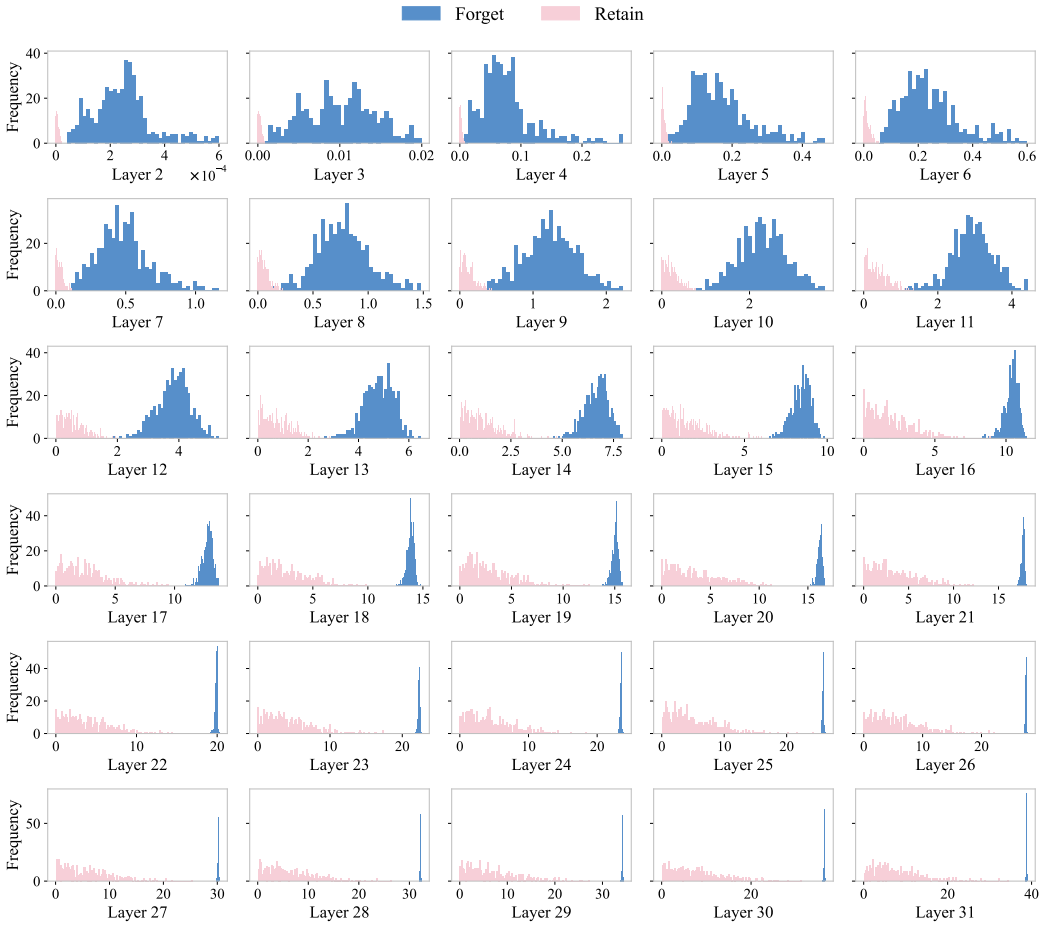


Figure 11: Layer-wise distributions of L2 norms for forget (blue) and retain (pink) steering vectors across Layers 2–31 of LLaVA-1.5-7B.

Table 10: Comparison of linear and MLP implementations of $f(h)$ on MLLMU-Bench (10% Forget) using LLaVA-1.5-7B. Ours uses the default linear implementation, while Ours-MLP employs a two-layer MLP. Results are reported on the forget (Fgt), test (Test), retain (Ret), and celebrity (Cele) sets. ↓ and ↑ denote lower-is-better and higher-is-better metrics, respectively.

Models	Classification Accuracy (%)				Generation: Rouge Score				Cloze: Accuracy (%)			
	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑	Fgt ↓	Test ↓	Ret ↑	Cele ↑
LLaVA-1.5-7B (10% Forget)-VQA												
Vanilla	43.60	41.20	45.26	47.00	0.591	0.339	0.591	0.312	57.00	15.00	55.11	8.17
Ours-MLP	29.20	32.40	38.62	42.19	0.363	0.265	0.414	0.311	19.00	8.00	32.00	6.54
Ours	**2.00**	**27.20**	**45.08**	**46.48**	**0.176**	**0.202**	**0.594**	**0.314**	**5.00**	**6.00**	**54.33**	**7.84**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state that the paper proposes MLLMERASER, an input-aware, training-free framework for test-time unlearning in MLLMs through activation steering. The claimed contributions are supported by the method design and experimental results on LLaVA-1.5 and Qwen-2.5-VL.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated limitation section discussing the reliance on adversarially perturbed images and hand-crafted prompts, as well as the current focus on image–text privacy-sensitive benchmarks rather than broader settings such as copyright removal, video-language models, or embodied agents.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provides the mathematical formulation of the input-aware steering mechanism and refers to Appendix for the proof related to the null-space projection used in the closed-form solution.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper specifies the evaluated backbones, benchmark, datasets, tasks, metrics, baselines, and implementation details. Appendix provides further details on benchmark construction, baselines, evaluation metrics, and implementation settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the code in the supplementary material, together with instructions for reproducing the main experimental results. The experiments are based on publicly available MLLM backbones and benchmarks, including LLaVA-1.5, Qwen-2.5-VL, and MLLMU-Bench.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the evaluated models, benchmark, forget/retain/test/celebrity splits, tasks, metrics, and baseline settings. Appendix further reports implementation details, including LoRA usage for baselines, steering strength, regularization parameters, perturbation radius.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The paper reports results across multiple tasks, datasets, models, and forget ratios, but it does not report error bars, confidence intervals, or statistical significance tests. Repeating all MLLM unlearning experiments multiple times would be computationally expensive.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports training and inference time comparisons and provides GPU memory usage in Appendix. The implementation details state that experiments are conducted on NVIDIA A800 GPUs with 80GB memory.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research focuses on improving privacy-preserving and safety-oriented MLLM unlearning. The experiments use existing public models and benchmarks, and the method is evaluated for reducing the generation of designated sensitive or unsafe information.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses the motivation of MLLM unlearning for privacy protection, misuse mitigation, and reliability. The limitation section also acknowledges the current scope of the evaluation and the need to extend the method to broader unlearning scenarios.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new high-risk pretrained model, image generator, or scraped dataset. It proposes a test-time intervention method and evaluates it on existing MLLM backbones and benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original sources for the models, datasets, benchmarks, and baselines used in the experiments, including LLaVA-1.5, Qwen-2.5-VL, MLLMU-Bench, ImageNet, and prior MLLM unlearning methods.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce a new dataset or pretrained model asset. The main contribution is a method for test-time MLLM unlearning.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, user studies, or participant data collection.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for writing polishing.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.