
Visual-Guided Key-Token Regularization for Multimodal Large Language Model Unlearning

Chengyi Cai¹, Zesheng Ye¹, Peike Li², Bo Han³, Jianzhong Qi¹, Feng Liu¹
¹The University of Melbourne ²Google Research ³Hong Kong Baptist University
fengliu.ml@gmail.com

Abstract

Unlearning in Multimodal Large Language Models (MLLMs) prevents the model from revealing private information when queried about target images. Existing MLLM unlearning methods largely adopt approaches developed for LLMs. They treat all answer tokens uniformly, disregarding their varying importance in the unlearning process. Moreover, these methods focus exclusively on the language modality, disregarding visual cues that indicate key tokens in answers. In this paper, after formulating the problem of unlearning in multimodal question answering for MLLMs, we propose *Visual-Guided Key-Token Regularization* (ViKeR). We leverage irrelevant visual inputs to predict ideal post-unlearning token-level distributions and use these distributions to regularize the unlearning process, thereby prioritizing key tokens. Further, we define key tokens in unlearning via *information entropy* and discuss ViKeR’s effectiveness through token-level *gradient reweighting*, which amplifies updates on key tokens. Experiments on MLLMU and CLEAR benchmarks demonstrate that our method effectively performs unlearning while mitigating forgetting and maintaining response coherence.

1 Introduction

Multimodal Large Language Models (MLLMs) have shown remarkable capabilities in vision-language understanding and generation by merging visual encoders and Large Language Model (LLM) backbones [44, 43, 38]. Nevertheless, their tendency to memorize training data brings increasing concerns about privacy leakage and copyright violations [16, 30]. To address these concerns, machine unlearning [9] in MLLMs seeks to prevent the model from disclosing private information when queried about target persons [11, 24], given the target set of visual-question-answer triples (referred to as the *forget set*). Meanwhile, non-target information and the model’s capability to produce coherent answers should be preserved, as shown in Figure 1.

Research on MLLM unlearning is still at an early stage, and many existing approaches [24] are adapted from well-established LLM unlearning methods [42, 45]. These methods typically assign a well-designed unlearning loss to the ground-truth answers corresponding to visual-question pairs in the forget set, discouraging the model from producing them. Such losses treat all tokens in the answer equally, which may instead cause less critical tokens that form the basic structure of a response without sensitive information (i.e., normal tokens) to be forgotten. As a result, the MLLM may lose its ability to generate coherent responses. Considering *gradient ascent* (GA) [42] as an example, the probability distributions of the predicted answer tokens for a visual-question pair (from the forget set of MLLMU [24]) before and after unlearning are shown in Figure 2(a) and Figure 2(b). Gray tokens indicate normal tokens, while red tokens indicate key tokens carrying critical information. It is observed that normal tokens are forgotten, leading to unreasonable token predictions, such as



Figure 1: Problem formulation for unlearning. Given the forget set (i.e., target visual-question-answer triples to be unlearned), the MLLM is expected to forget the targeted content while preserving other knowledge and generating coherent responses.

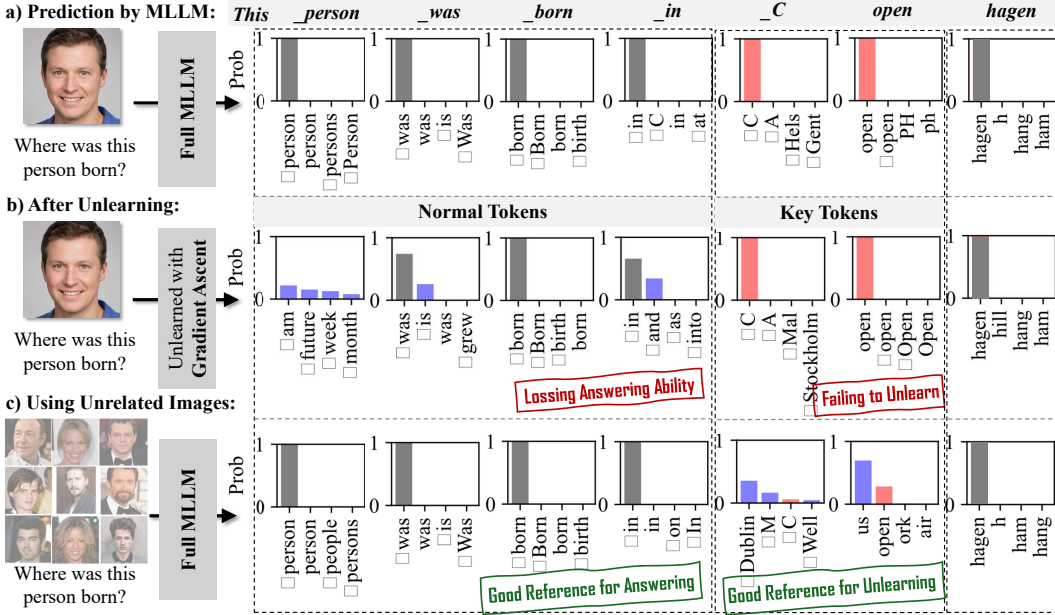


Figure 2: Probability distributions of the predicted answer tokens for a visual-question pair to be unlearned. a) Output of the full MLLM (i.e., LLM before unlearning). Gray indicates normal tokens, while red indicates key tokens. b) Output of MLLM after unlearning with GA, where normal tokens are forgotten instead of key tokens. c) Averaged output of the full MLLM, using irrelevant reference images as inputs, which approaches the ideal distribution for unlearning, and can be considered as a good reference for the unlearning process.

‘This am’. However, key tokens such as ‘_C’ and ‘open’ that form ‘Copenhagen’ and reveal private information tend to be kept. This highlights the drawback of ignoring token-level importance. However, automatically identifying key tokens in a sentence for unlearning remains a challenging problem. We observe that visual inputs can be effective in guiding key-token identification, an aspect that has been largely neglected by previous MLLM unlearning methods that focus solely on the language modality. We randomly choose 9 images from Celeb [26], which are irrelevant to the forget set, to replace the image in Figure 2(a). These images, along with the same question, are fed into the full MLLM (i.e., the model before unlearning with full information), and the averaged token-level distributions are shown in Figure 2(c). It is observed that the predictions for normal tokens resemble those in Figure 2(a), while key tokens no longer convey private information, reflecting the ideal token probabilities after unlearning. Therefore, these distributions can serve as a reference to regularize the unlearning process.

Section 3 formalizes the problem setting of MLLM unlearning, focusing on forgetting visual-question-answer triples. We further clarify that the objective is to remove the forget set while preserving non-target knowledge and MLLM’s ability to generate coherent responses, as shown in Figure 1.

In Section 4, inspired by the observation in Figure 2, we propose a *Visual-Guided Key-Token Regularization* (ViKeR) method for MLLM unlearning. ViKeR consists of two steps: visual-guided token distribution estimation and token-level regularization. Firstly, ideal token distributions are estimated using irrelevant visual inputs. Secondly, the estimated distributions are used to regularize the MLLM unlearning loss, which enables unlearning while preserving retention and response coherence.

In Section 5, we use *information entropy* to formally define key tokens. We then provide a discussion of ViKeR’s effectiveness by exploring its token-level *gradient reweighting*, in comparison with GA.

In Section 6, experimental results on the MLLMU [24] and CLEAR [6] benchmarks with LLaVA-7B [23] demonstrate the effectiveness of ViKeR for MLLM unlearning. While achieving comparable unlearning performance, ViKeR outperforms existing unlearning methods in preserving non-target knowledge and maintaining output readability. The ablation studies, hyperparameter analyses, investigations of regularizers and reference images comprehensively investigate its performance and validate the soundness of the design.

In conclusion, both theoretical analysis and experimental results demonstrate the effectiveness of ViKeR. It offers new perspectives on improving MLLM unlearning through the use of visual modality and token-level importance.

2 Related Works

LLM Unlearning. With the development of LLMs and the growing need for privacy protection [8, 41, 12], LLM unlearning has gradually emerged as a mature research direction. Recent studies cover a wide range of aspects, including LLM unlearning benchmarks [27, 34, 20], evaluation of methods [36, 35], unified frameworks [7], and methodological improvements [13, 29, 15, 37, 40].

One of the most well-known approaches is the *gradient ascent* (GA) [42], which uses the negative log-likelihood typically employed in standard LLM training, but with the sign inverted as the loss function, allowing the model to forget the target data. In addition, *direct preference optimization* (DPO) [31] has also been applied to unlearning by treating responses such as “I don’t know” as preferred answers, encouraging the model to refuse to answer queries about private data. *Negative preference optimization* (NPO) [45], adapts DPO by removing the preferred-answer component, satisfying cases when only non-preferred responses are available.

MLLM Unlearning. Machine unlearning [1, 2, 9] has gradually been applied to MLLMs to remove harmful information [5] or private data [6, 24]. Currently, MLLM unlearning lacks a unified problem formulation: some studies focus on multi-person scenarios [4], some on single images [18], some on the removal of harmful content [5], and others on the elimination of visual modality-related answers while retaining language information [11]. More discussion and evaluation involve hallucination mitigation [39], safety alignment [3], and attacks [46].

In this work, we focus on unlearning in multimodal question answering. To better align with practical applications, we restrict access during the unlearning stage to only the forget set, which is significantly smaller than the full dataset. Current MLLM unlearning methods sometimes involve internal architectural modifications, such as neuron pruning [25], neuron path editing [19], or selective fine-tuning of specific weights [24]. However, considering architecture-agnostic unlearning loss design under our problem formulation, most available methods largely follow approaches from LLM unlearning [24], often neglecting the varying importance of tokens and potential visual guidance.

3 Preliminary

Token-Level Probabilities and MLLM Training Loss. We consider an MLLM parameterized by θ , where the output of the image encoder is mapped by a projector into visual tokens and concatenated with LLM’s text tokens [23]. Let a visual-question-answer triple s consist of an image I , a question x , and a ground-truth answer y , i.e., $s = (I, x, y)$. The answer y is represented as a sequence of tokens $y = [y_1, y_2, \dots, y_{|y|}]$. Let $y_{<i}$ denote the prefix $[y_1, \dots, y_{i-1}]$ of the answer y , the probability that the predicted i -th token \hat{y}_i takes a specific value v is defined as

$$p_\theta(v|I, x, i) \triangleq p(\hat{y}_i = v|I, x, y_{<i}; \theta), \quad (1)$$

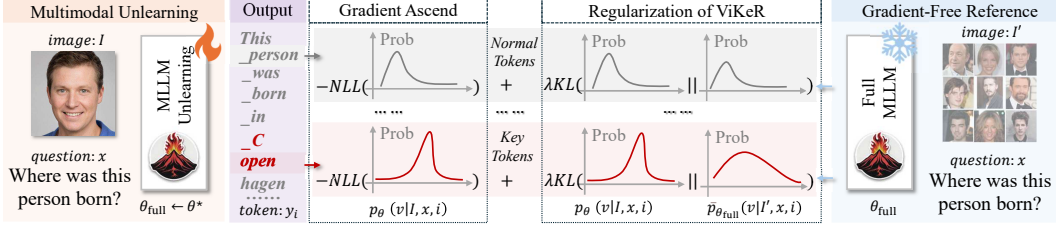


Figure 3: ViKeR Pipeline. The left half shows GA, while the right depicts modifications of ViKeR. ViKeR has two stages: (1) *visual-guided token distribution estimation* (in blue), where irrelevant visual inputs are fed into the pre-unlearning gradient-free full model to estimate ideal token distributions, and (2) *token-level regularization* (in orange), using the predicted distribution to regularize GA.

which is the output of the auto-regressive MLLM on each step for predicting the next token. Then the probability of answer y is $p(y|I, x; \theta) = \prod_{i=1}^{|y|} p_{\theta}(y_i|I, x, i)$, i.e., the product of probabilities that the i -th predicted token \hat{y}_i matches the ground truth token y_i . For a visual-question-answering task with training set $\mathcal{D}_{\text{full}} = \{s^1, s^2, \dots, s^{|\mathcal{D}|}\}$, the MLLM can be fine-tuned to obtain the full model θ_{full} with loss

$$\mathcal{L}_{\text{NLL}}(\mathcal{D}_{\text{full}}; \theta) = -\frac{1}{|\mathcal{D}|} \sum_{s \in \mathcal{D}_{\text{full}}} \log p(y|I, x; \theta), \quad (2)$$

where $L_{\text{NLL}}(\cdot; \cdot)$ denotes the negative log-likelihood.

Problem Formulation for MLLM Unlearning. Similar to LLM unlearning, we define the forget set as $\mathcal{D}_f = \{s_f^1, s_f^2, \dots, s_f^{|\mathcal{D}_f|}\} \subset \mathcal{D}_{\text{full}}$, consisting of visual-question-answer triples to be unlearned from θ_{full} . The size of these datasets typically satisfies $|\mathcal{D}_f| \ll |\mathcal{D}_{\text{full}}|$. During the unlearning phase, only θ_{full} and \mathcal{D}_f are available, which reflects real-world usage scenarios. Then the goal of unlearning is to derive a model θ^* that simultaneously satisfies:

- (*Forgetting*). For any $(I, x, y) \in \mathcal{D}_f$, θ^* no longer outputs the correct answer y when queried with the visual-question pair (I, x) .
- (*Retention*). For any $(I, x, y) \in \mathcal{D}_{\text{full}} \setminus \mathcal{D}_f$, the output of θ^* remains unchanged.
- (*Coherence*). For any $(I, x, y) \in \mathcal{D}_{\text{full}}$, the model θ^* preserves its ability to generate coherent, non-gibberish outputs for inputs (I, x) .

Gradient Ascent (GA). An intuitive approach is to reuse the negative log-likelihood in Eq. (2) on \mathcal{D}_f , but with the sign flipped, and use it as the unlearning loss:

$$\mathcal{L}_{\text{GA}}(\mathcal{D}_f; \theta) = -\mathcal{L}_{\text{NLL}}(\mathcal{D}_f; \theta). \quad (3)$$

Substituting Eq. (1) and Eq. (2) into Eq. (3), we can express the gradient of GA loss in terms of individual tokens:

$$\nabla_{\theta} \mathcal{L}_{\text{GA}}(\mathcal{D}_f; \theta) = \frac{1}{|\mathcal{D}_f|} \sum_{(I, x, y) \in \mathcal{D}_f} \sum_{i=1}^{|y|} \nabla_{\theta} \mathcal{L}_{\text{GA}}(y_i; \theta, y, i),$$

where $\nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i) \triangleq \nabla_{\theta} \log p_{\theta}(v|I, x, i)$ is defined as the token-level gradient of the i -th token in answer y being v . See detailed derivation in Appendix B.1.

Limitations of GA in MLLM unlearning. First, GA simply averages gradients over \mathcal{D}_f and assigns equal weight to all answer tokens, which leads to over-forgetting normal tokens and disrupted coherence, as illustrated in Figure 2(b). Second, the GA loss in Eq. (3) has the same form for MLLMs and text-only LLMs, treating the multimodal input (I, x) as a flat token sequence and ignoring visual structure. Consequently, GA fails to leverage visual cues to distinguish identity-dependent tokens (i.e., key tokens) from those that should remain stable (i.e., normal tokens).

4 Visual-Guided Key-Token Regularization

Method Overview. The pipeline of ViKeR, illustrated in Figure 3, consists of two components: *visual-guided token distribution estimation* and *token-level regularization*. The first part leverages

irrelevant visual inputs (i.e., images of individuals not included in \mathcal{D}_f) fed into the pre-unlearning full model to predict the ideal post-unlearning token distribution (as observed in Figure 2(c)) in a gradient-free manner. The second component uses the predicted ideal distribution to regularize the GA unlearning process.

Visual-Guided Token Distribution Estimation. Shown in Figure 2(c), a set of unlabeled images of irrelevant individuals can be used to estimate the ideal post-unlearning token distribution for each generated answer token.

Assuming that k images of irrelevant individuals from a reference set $\mathcal{I}' = \{I'_1, I'_2, \dots, I'_k\}$ are used, the ideal distribution of the i -th token in an answer y is estimated as:

$$\hat{\mathcal{R}}_i^y(v) = \bar{p}_{\theta_{\text{full}}}(v|\mathcal{I}', x, i) \triangleq \frac{1}{k} \sum_{j=1}^k p_{\theta_{\text{full}}}(v|I'_j, x, i), \quad (4)$$

where $\hat{\mathcal{R}}_i^y$ is the distribution over tokens $v \in \mathcal{V}$, with \mathcal{V} being the complete token set, and $p_{\theta_{\text{full}}}(v|\cdot)$ is the probability of token v as defined in Eq. (1), given the full model θ_{full} , a visual-question-answer triple (I'_j, x, y) and index i .

It can be noticed that computing $\hat{\mathcal{R}}_i^y$ only requires forward passes through θ_{full} , being a gradient-free inference process. Therefore, it incurs minimal computational overhead. The hyperparameter k is discussed in Section 6.

Token-Level Regularization. Given an MLLM model θ , a visual-question-answer triple $(I, x, y) \in \mathcal{D}_f$ and the token position i , the predicted distribution of the i -th token in y is

$$\hat{\mathcal{Q}}_i^y(v) = p_{\theta}(v|I, x, i),$$

as defined in Eq. (1). $\hat{\mathcal{Q}}_i^y$ is the distribution over $v \in \mathcal{V}$.

After ideal unlearning, we would like $\hat{\mathcal{Q}}_i^y$ to approximate $\hat{\mathcal{R}}_i^y$ for all $(I, x, y) \in \mathcal{D}_f$ and positions i . The intuition is that unlearning is achieved when, for any $(I, x, y) \in \mathcal{D}_f$ fed into the MLLM, the resulting token distributions resemble the expected distributions obtained by inputting the same question x and identity-agnostic individuals. It is supported by the observation in Figure 2(c): across reference images from \mathcal{I}' , the averaged distributions $\hat{\mathcal{R}}_i^y$ do not exhibit individual-specific peaks for *key tokens* associated with private attributes, while remaining sharp for *normal tokens*.

Therefore, the discrepancy between $\hat{\mathcal{Q}}_i^y$ and $\hat{\mathcal{R}}_i^y$ can be used to regularize the GA unlearning process. ViKeR employs the widely-used Kullback-Leibler (KL) divergence [17] for regularization, so the unlearning loss can be formulated as:

$$\mathcal{L}_{\text{ViKeR}}(\mathcal{D}_f; \theta) = -\mathcal{L}_{\text{NLL}}(\mathcal{D}_f; \theta) + \frac{\lambda}{|\mathcal{D}_f|} \cdot \sum_{(I, x, y) \in \mathcal{D}_f} \sum_{i=1}^{|y|} \text{KL}(\hat{\mathcal{R}}_i^y \parallel \hat{\mathcal{Q}}_i^y),$$

where $\text{KL}(\cdot \parallel \cdot)$ is the KL divergence, with a balance hyperparameter $\lambda \in [0, 1]$, which is further discussed in Section 6.

5 Understanding ViKeR Through Token-Level Gradient Reweighting

Let \mathcal{R}_i^y denote the ideal post-unlearning token distribution at position i in answer y , which is approximated in practice by $\hat{\mathcal{R}}_i^y$ defined in Eq. (4). Then normal and key tokens in the ground truth answer y are defined as follows.

Definition 1. (*Normal Tokens.*) Let $f_{\text{nr}}(\cdot)$ be an indicator function that marks whether a token is a normal token. We define a token y_i as a normal token if its estimated ideal distribution \mathcal{R}_i^y is sufficiently close to the Dirac distribution δ_{y_i} (i.e., a one-hot encoding at y_i). Formally, this is captured by the indicator function:

$$f_{\text{nr}}(y_i) \triangleq \mathbb{I}\{\mathcal{R}_i^y(y_i) \geq \tau\}$$

where $\tau \in (0, 1]$ is a confidence threshold satisfying $\tau \rightarrow 1$.

For example, ‘_person’, ‘_was’, ‘_born’, ‘_in’, ‘hagen’ in Figure 2 can be normal tokens.

Proposition 1. Let $H(\cdot)$ denote the information entropy [33]. Then for a normal token y_i :

$$f_{\text{nr1}}(y_i) = 1 \Rightarrow H(\mathcal{R}_i^{y_i}) \rightarrow 0,$$

proved in Appendix B.2. The ideal post-unlearning distribution of a normal token has (approximately) zero entropy.

Accordingly, key tokens can be defined distinctly:

Definition 2. (Key Tokens.) Let $f_{\text{key}}(\cdot)$ be an indicator function that marks whether a token is a key token. Then

$$f_{\text{key}}(y_i) \triangleq \mathbb{I}\{H(\mathcal{R}_i^{y_i}) \geq \epsilon\},$$

where ϵ denotes a user-specified threshold and is independent of the subsequent propositions.

For example, ‘_C’, ‘open’ in Figure 2 can be key tokens.

By definition, key tokens and normal tokens should be disjoint and do not cover all tokens, and they are treated with different importance during unlearning. Next, we differentiate the ViKeR loss in Eq. (5) to analyze its token-level gradients and compare them to those of GA.

Proposition 2. Recall the token-level GA gradient $\nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i)$ defined in Section 3. If y_i is a normal token, then its token-level gradient under ViKeR can be written as a reweighted version of the GA gradient:

$$\nabla_{\theta} \mathcal{L}_{\text{ViKeR}}(v; \theta, y, i) = \begin{cases} (1 - \lambda) \cdot \nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i) & \text{if } v = y_i \\ 0 \cdot \nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i) & \text{if } v \neq y_i \end{cases},$$

where v is the value of token variable in i -th location of answer y . Detailed proof is in Appendix B.3.

Thus, for normal tokens, the unlearning gradients are scaled by a factor $1 - \lambda$ (or set to zero for $v \neq y_i$), which reduces the magnitude of updates on these tokens and helps preserve the model’s ability to generate them.

Proposition 3. Using the definition in Eq. (4), the token-level gradient under ViKeR can be written, for any token (with detailed proof in Appendix B.3), as:

$$\nabla_{\theta} \mathcal{L}_{\text{ViKeR}}(v; \theta, y, i) = (\mathbb{I}\{v = y_i\} - \lambda \cdot \bar{p}_{\theta_{\text{full}}}(v|\mathcal{I}', x, i)) \cdot \nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i),$$

with Proposition 2 following as a special case.

Key tokens are given more priority compared with normal ones, as the reweighting scale is larger than $1 - \lambda$ (of normal tokens) when $v = y_i$. For other candidate tokens $v \neq y_i$, the gradients are reweighted in the opposite direction to unlearning, which leads θ to favor other candidate tokens, also contributing to forgetting the target y_i .

6 Experiments

Benchmarks and Baselines. Experiments are conducted on the MLLMU [24] and CLEAR [6] benchmarks using LLaVA-7B [23] with LoRA [10] fine-tuning, following [24]. Benchmark details and evaluation metrics are provided in Appendix A.1 and Appendix A.2. Baselines include GA [42], NPO [45], and IdkPO (i.e., using ‘I don’t know ...’ as preferred samples in DPO [31]). All methods share the same training hyperparameters for a fair comparison. Baseline descriptions and implementation details are given in Appendix A.3 and Appendix A.4.

Comparison Results. The results on MLLMU are shown in Table 1. Methods are evaluated along ‘Forget’, ‘Generalization’, ‘Retain’, and ‘Real’. ‘Forget’ reports results on the forget set, while ‘Generalization’ evaluates the set from different views; both are measured using classification accuracy (ACC) for unlearning. ‘Retain’ corresponds to synthetic persons to be retained, and ‘Real’ refers to real persons; both are evaluated using ROUGE [21] and BLEU [28] to measure information retention. GIB [14] is used to assess whether the generated responses are gibberish. ‘Origin’ and ‘Full’ denote the MLLMs before and after training on $\mathcal{D}_{\text{full}}$, respectively, and serve as references for unlearning and retaining performance. It can be observed from Table 1 that GA and NPO show a clear degradation in GIB, while IdkPO substantially harms retention, resulting in low ROUGE and BLEU scores. In contrast, ViKeR achieves forgetting accuracy close to the original MLLM while maintaining the best ROUGE, BLEU, and GIB performance. Its advantage becomes more pronounced in the task with more forgetting data (i.e., 15%), yielding ROUGE gains of +41.4% / +24.1% and BLEU gains of +21.1% / +9.6% on ‘Retain’ and ‘Real’, respectively.

Table 1: Results of different methods on MLLMU, with the best in **bold** and ours highlighted. All reported values are percentages (%).

10% Task	Forget		Generalization		Retain			Real		
	Acc	Gib↑	Acc	Gib↑	Rouge↑	BLEU↑	Gib↑	Rouge↑	BLEU↑	Gib↑
Origin	25.2	-	30.8	-	-	-	-	-	-	-
Full	-	-	-	-	49.7	23.1	-	22.6	6.1	-
GA	25.3±2.9	83.7±2.1	30.1±2.5	78.5±2.7	20.6±2.9	3.0±0.7	81.3±3.5	17.5±2.1	2.4±0.6	87.8±2.6
NPO	30.8±1.7	89.9±0.9	33.1±1.4	80.9±1.9	29.7±1.1	5.7±0.5	89.9±0.8	23.2±0.6	4.6±0.2	91.2±0.5
IdkPO	36.9±1.3	94.4±0.2	36.5±0.2	73.2±1.4	5.7±0.5	1.3±0.2	91.7±0.2	4.2±0.1	0.6±0.0	86.3±0.1
ViKeR	30.4±3.6	95.4±0.3	30.1±1.9	88.2±3.3	32.4±1.4	6.9±0.5	94.6±0.4	24.6±1.1	4.7±0.4	93.7±0.2

15% Task	Forget		Generalization		Retain			Real		
	Acc	Gib↑	Acc	Gib↑	Rouge↑	BLEU↑	Gib↑	Rouge↑	BLEU↑	Gib↑
Origin	28.0	-	29.6	-	-	-	-	-	-	-
Full	-	-	-	-	50.0	24.2	-	22.6	6.1	-
GA	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.1±0.0	0.0±0.0	0.0±0.0	0.1±0.0	0.0±0.0	0.0±0.0
NPO	7.1±1.3	50.8±6.8	12.0±1.1	57.9±2.0	11.3±0.7	0.9±0.1	51.4±5.0	9.1±0.6	0.7±0.1	51.0±3.8
IdkPO	34.7±0.9	92.1±0.6	37.2±0.3	82.2±1.9	5.2±0.2	0.6±0.0	93.6±0.6	4.0±0.2	0.4±0.0	91.3±1.3
ViKeR	32.0±3.1	93.2±0.7	32.4±3.4	83.1±0.8	52.7±1.8	22.0±1.7	94.6±0.0	33.2±0.6	10.3±0.2	92.0±0.7

Table 2: Results on the CLEAR dataset (% , best is in **bold** and ours is highlighted).

10% Task	Forget		Retain		Real QA	
	Rec↓	Gib↑	Rec↓	Gib↑	Acc (Aut)↑	Acc (Wor)↑
GA	0.00	11.60	0.00	11.27	50.55	47.77
	±0.00	±9.92	±0.00	±10.27	±1.74	±0.13
NPO	0.18	94.16	0.80	93.31	69.98	48.41
	±0.12	±0.66	±0.17	±0.70	±0.62	±0.13
IdkPO	0.00	93.33	0.65	93.35	75.28	49.95
	±0.00	±0.32	±0.05	±0.49	±0.31	±0.13
ViKeR	0.62	92.98	4.21	93.42	76.60	48.96
	±0.12	±0.88	±1.49	±0.80	±0.31	±0.34

Results on the CLEAR dataset are shown in Table 2, evaluated from three perspectives. ‘Forget’ and ‘Retain’ measure unlearning and retaining via identity recognition accuracy (REC) with GIB used in gibberish-judging, while ‘Real QA’ assesses answer accuracy (ACC) on real authors (AUT) and world (WOR) images. It can be observed that ViKeR maintains answer accuracy on real-world questions and fluent language generation (i.e., a high GIB) on par with SOTA. Moreover, it substantially improves retention (+3.41%) while minimally affecting unlearning (+0.48%). Samples of visual-question-answer triples from MLLMU

and CLEAR are provided in Appendix C.1, showing responses after unlearning by different methods.

Visualization Results of Token Distribution. Figure 4 illustrates an example from the forget set, including the image, the associated question, and the answer. We visualize the predicted token distributions after unlearning with different methods, using the normal token ‘_person’ (gray bars) and the key token ‘_civil’ (red bars) as representative cases. The results show that GA significantly perturbs the distribution of normal tokens, leading to responses that drift away from the original question. While NPO maintains the ability to continue answering the question, it does not fully suppress the probability mass of the key token, and private information may still be leaked given the preceding context. IdkPO exhibits similar issues, including distortion of normal token predictions and failure in unlearning the key token. In contrast, our ViKeR effectively eliminates the key token while preserving coherent normal answers. More visualization results are illustrated in Appendix C.2.

Ablation Studies. Table 3 reports the ablation studies. ‘W/o Reg’ denotes removing the second term in Eq (5), i.e., the regularization term. ‘W/o GA’ indicates removing the first term, i.e., the positive log-likelihood term. ‘W/o Vis’ refers to removing the visual guidance and applying regularization solely based on the image’s initial token distribution, i.e., using $p_{full}(v|I, x, i)$ as $\hat{\mathcal{R}}_i^y$ in Eq (4).

It can be observed that ‘w/o Reg’ degenerates to GA, leading to pronounced forgetting as well as a noticeable degradation in response coherence. This is evidenced by the lower ROUGE and BLEU scores on the ‘Retain’ and ‘Real’ sets, together with a reduced GIB score. In contrast, ‘w/o GA’ weakens the unlearning capability, resulting in an unexpected increase in ACC on the ‘Forget’ and ‘Generalization’ sets. These results indicate that both terms in Eq (5) are indispensable, validating the design of the token-level regularization. Although ‘w/o Vis’ relatively preserves response coherence, it still exhibits a significant drop in retaining performance, suggesting that our visual-guided token distribution estimation is likewise necessary. Please refer to Appendix C.3 for more results.

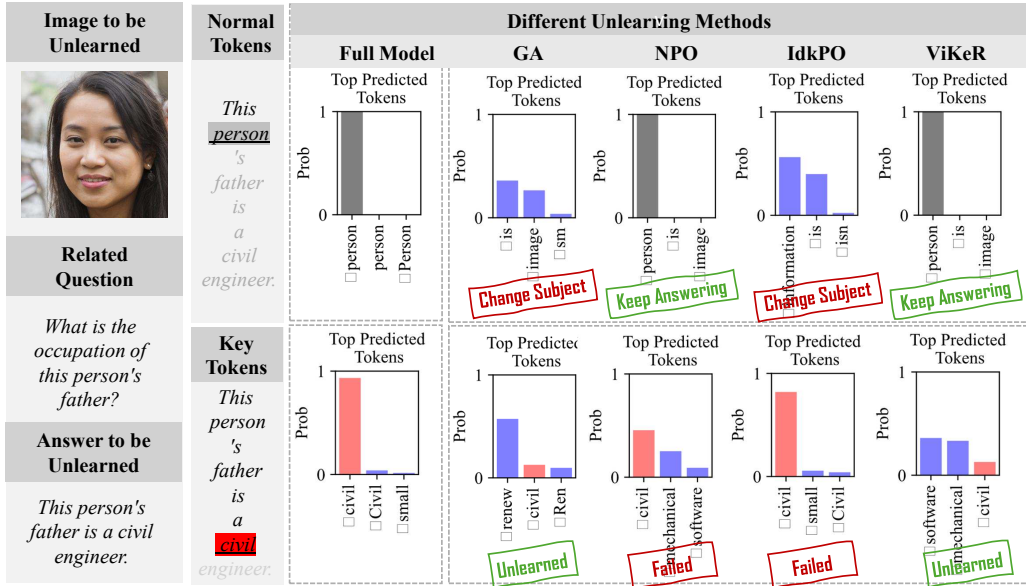


Figure 4: Visualization results of token distribution after unlearning with different methods. Compared to other methods, ViKeR preserves the prediction of normal tokens (such as ‘_person’) while successfully unlearning key tokens (such as ‘_civil’).

Table 3: Results of ablation studies on MLLMU, with the best in **bold** and ours highlighted (%).

10% Task	Forget		Generalization		Retain			Real		
	Acc	Gib↑	Acc	Gib↑	Rouge↑	BLEU↑	Gib↑	Rouge↑	BLEU↑	Gib↑
Origin	25.2	-	30.8	-	-	-	-	-	-	-
Full	-	-	-	-	49.7	23.1	-	22.6	6.1	-
w/o Reg	25.3 ±2.9	83.7±2.1	30.1 ±2.5	78.5±2.7	20.6±2.9	3.0±0.7	81.3±3.5	17.5±2.1	2.4±0.6	87.8±2.6
w/o GA	44.9±0.2	89.5±1.0	39.5±0.5	73.7±1.5	41.8±1.0	16.3±0.5	87.3±0.8	20.1±0.4	5.8±0.1	74.2±1.0
w/o Vis	26.7±3.8	90.1±0.4	30.9±2.6	85.3±0.3	28.2±1.9	5.0±0.6	90.1±0.5	21.5±1.6	3.7±0.7	91.7±0.2
ViKeR	30.4±3.6	95.4 ±0.3	30.1 ±1.9	88.2 ±3.3	32.4 ±1.4	6.9 ±0.5	94.6 ±0.4	24.6 ±1.1	4.7 ±0.4	93.7 ±0.2

Hyperparameter Analysis. Figure 5 shows the effects of the regularization strength λ and the number of irrelevant visual inputs k used to estimate \hat{R}_i^y . The first row reports GIB, measuring the coherence of MLLM responses after unlearning, while the second row shows unlearning and retaining performance, evaluated by ACC and ROUGE, respectively.

As the regularization strength increases, retaining performance improves, but unlearning effectiveness degrades. When λ is relatively large, the GIB score decreases and becomes unstable, with larger variance. We set $\lambda = 0.5$ for the 15% tasks as a balanced choice. In contrast, the number of irrelevant visual inputs k has a smaller impact on the results. With fewer reference images, the performance is unstable and shows higher variance. When $k \geq 5$, the metrics become stable. We adopt $k = 5$ in our experiments. Detailed results are included in Appendix C.4.

Results of Different Regularization. ViKeR employs KL divergence to regularize the distance between \hat{R}_i^y and \hat{Q}_i^y , while other similarity measures can also be used as regularizers. Figure 6 presents the results using cosine similarity (CoS) [22] and Jensen–Shannon divergence (JSD) [32].

Compared to the setting without regularization, CoS substantially improves language coherence, but shows limited gains in unlearning and retaining performance. The results of JSD are nearly identical to those of KL divergence. However, from a computational perspective, JSD is approximately twice as expensive as KL divergence. Therefore, adopting KL divergence as the regularization term is a more reasonable choice. Detailed results are included in Appendix C.5.

Results of Different Visual References. To study the impact of reference image set \mathcal{I}' , we further experiment with images from the forget set (‘forget’), the retain set (‘retain’), and other categories, including cats or dogs (‘pets’), real-world scenes (‘scene’), and different textures (‘pattern’), aside from the irrelevant celebrities (‘peoples’) we use in ViKeR. The results are in Figure 7 and Appendix C.6.

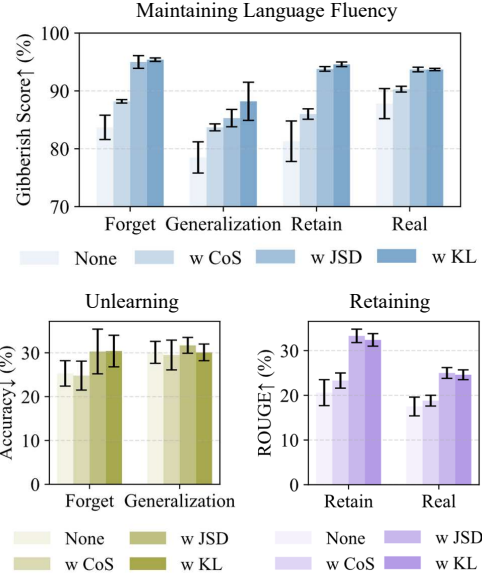
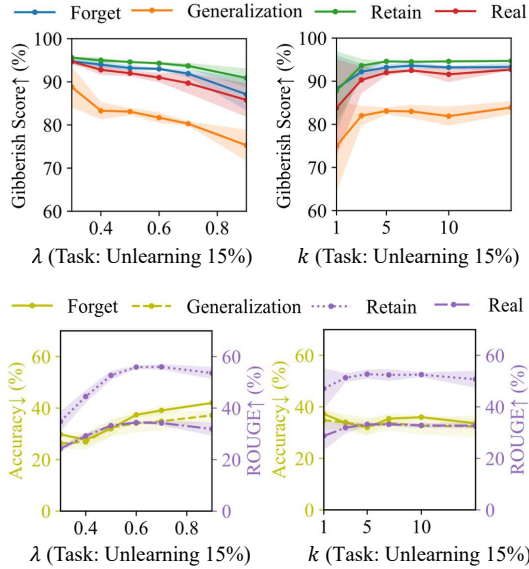


Figure 5: Hyperparameter analysis of λ and k .

Figure 6: Different regularizers on 10% Task.

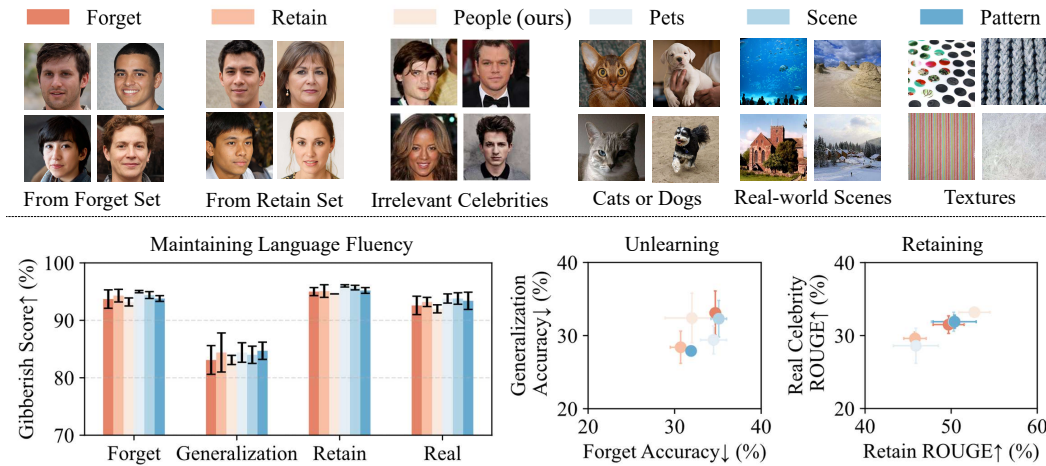


Figure 7: Results of ViKeR on MLLMU (15% Task) when using different kinds of images as the reference set \mathcal{I}' , other than the celebrities.

Types of references have little effect on GIB, indicating that language coherence is preserved regardless of \mathcal{I}' . However, unlearning and retaining performance vary. Using ‘pattern’ and ‘retain’ images yields the best unlearning performance, likely due to their relatively consistent internal distributions and large discrepancy from the forget set. In contrast, ‘forget’ and ‘scene’ images are the worst, as ‘forget’ images contain privacy information to unlearn, and ‘scene’ images exhibit high diversity, both of which hinder accurate ideal distribution estimation. For retaining performance, ‘retain’ and ‘pet’ images lead to noticeable degradation, possibly because their estimated distributions contain task-related or human-like information that is also suppressed during unlearning. In contrast, using ‘people’ images achieves a better balance between unlearning and retaining.

7 Conclusion

We propose ViKeR for MLLM unlearning with improved retention and response coherence. It leverages irrelevant visual inputs to estimate the ideal token distribution, which constrains the unlearning process via token-level regularization. Its efficacy is validated theoretically and empirically.

References

- [1] Lucas Bourtole, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE SP*, 2021.
- [2] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE SP*, 2015.
- [3] Trishna Chakraborty, Erfan Shayegani, Zikui Cai, Nael Abu-Ghazaleh, M Salman Asif, Yue Dong, Amit K Roy-Chowdhury, and Chengyu Song. Cross-modal safety alignment: Is textual unlearning all you need? *arXiv preprint arXiv:2406.02575*, 2024.
- [4] Haokun Chen, Jianing Li, Yao Zhang, Jinhe Bi, Yan Xia, Jindong Gu, and Volker Tresp. Auvic: Adversarial unlearning of visual concepts for multi-modal large language models. *arXiv preprint arXiv:2511.11299*, 2025.
- [5] 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. *arXiv preprint arXiv:2502.12520*, 2025.
- [6] Alexey Dontsov, Dmitrii Korzh, Alexey Zhavoronkin, Boris Mikheev, Denis Bobkov, Aibek Alanov, Oleg Rogov, Ivan Oseledets, and Elena Tutubalina. Clear: Character unlearning in textual and visual modalities. In *ACL (Findings)*, 2025.
- [7] Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zachary C Lipton, J Zico Kolter, and Pratyush Maini. Openunlearning: Accelerating llm unlearning via unified benchmarking of methods and metrics. In *NeurIPS*, 2025.
- [8] Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. Towards llm unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. In *ICML*, 2025.
- [9] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *NeurIPS*, 2019.
- [10] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022.
- [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)*, 2025.
- [12] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *ACL*, 2023.
- [13] Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. Soul: Unlocking the power of second-order optimization for llm unlearning. In *EMNLP*, 2024.
- [14] Madhur Jindal. Gibberish detector: High-accuracy text classification model, 2021.
- [15] Swanand Ravindra Kadhe, Farhan Ahmed, Dennis Wei, Nathalie Baracaldo, and Inkit Padhi. Split, unlearn, merge: Leveraging data attributes for more effective unlearning in llms. In *ICML workshop*, 2024.
- [16] Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. In *EMNLP*, 2023.
- [17] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 1951.

- [18] Jiaqi Li, Qianshan Wei, Chuanyi Zhang, Guilin Qi, Miaozen Du, Yongrui Chen, Sheng Bi, and Fan Liu. Single image unlearning: Efficient machine unlearning in multimodal large language models. In *NeurIPS*, 2024.
- [19] Kunhao Li, Wenhao Li, Di Wu, Lei Yang, Jun Bai, Ju Jia, and Jason Xue. Cross-modal unlearning via influential neuron path editing in multimodal large language models. *arXiv preprint arXiv:2511.06793*, 2025.
- [20] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.
- [21] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 2004.
- [22] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 2002.
- [23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2023.
- [24] 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*, 2025.
- [25] Zheyuan Liu, Guangyao Dou, Xiangchi Yuan, Chunhui Zhang, Zhaoxuan Tan, and Meng Jiang. Modality-aware neuron pruning for unlearning in multimodal large language models. *arXiv preprint arXiv:2502.15910*, 2025.
- [26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.
- [27] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.
- [28] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.
- [29] Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. In-context unlearning: Language models as few-shot unlearners. In *ICML*, 2024.
- [30] Renjie Pi, Tianyang Han, Jianshu Zhang, Yueqi Xie, Rui Pan, Qing Lian, Hanze Dong, Jipeng Zhang, and Tong Zhang. Mllm-protector: Ensuring mllm’s safety without hurting performance. *arXiv preprint arXiv:2401.02906*, 2024.
- [31] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 2023.
- [32] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 1975.
- [33] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 1948.
- [34] Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A Smith, and Chiyuan Zhang. Muse: Machine unlearning six-way evaluation for language models. In *ICLR*, 2025.
- [35] Qizhou Wang, Bo Han, Puning Yang, Jianing Zhu, Tongliang Liu, and Masashi Sugiyama. Towards effective evaluations and comparisons for llm unlearning methods. In *ICLR*, 2025.
- [36] Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saeyool Shin, Bo Han, and Kilian Q Weinberger. Rethinking llm unlearning objectives: A gradient perspective and go beyond. In *ICLR*, 2025.

- [37] Yue Wang, Qizhou Wang, Feng Liu, Wei Huang, Yali Du, Xiaojiang Du, and Bo Han. Gru: Mitigating the trade-off between unlearning and retention for large language models. In *ICML*, 2025.
- [38] Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S Yu. Multimodal large language models: A survey. In *International Conference on Big Data*, 2023.
- [39] 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. *arXiv preprint arXiv:2402.09801*, 2024.
- [40] Puning Yang, Qizhou Wang, Zhuo Huang, Tongliang Liu, Chengqi Zhang, and Bo Han. Exploring criteria of loss reweighting to enhance llm unlearning. In *ICML*, 2025.
- [41] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 2024.
- [42] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *NerulPS*, 2024.
- [43] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 2024.
- [44] Duzhen Zhang, Yahan Yu, Jiahua Dong, Chenxing Li, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. In *ACL (Findings)*, 2024.
- [45] Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024.
- [46] Xianren Zhang, Hui Liu, Delvin Ce Zhang, Xianfeng Tang, Qi He, Dongwon Lee, and Suhang Wang. Does multimodal large language model truly unlearn? stealthy mllm unlearning attack. *arXiv preprint arXiv:2506.17265*, 2025.

A More Preliminary Knowledge

A.1 Benchmark Information

MLLMU [24] is a dataset consisting of 500 fictional characters and 153 public figures. Each character is associated with at least seven visual-question-answer triples and seven text-only question-answer pairs. The evaluation protocol includes four aspects: *Forget*, *Generalization*, *Retain*, and *Real*, corresponding to characters to be unlearned, alternative images of the same characters, characters to be retained, and real-person question-answer pairs, respectively. Since this work focuses on vision-language tasks, we only use visual-question-answer triples for our experiments.

CLEAR [6] is a dataset consisting of 200 fictional characters and 3,700 images paired with question-answer annotations. The evaluation primarily considers four aspects: *Forget*, *Retain*, *Questions about real authors*, and *Questions about the real world*. For the *Forget* and *Retain* settings, we focus on person identification, where the question "What is the name of the person in the image" is used, along with ten different name-containing answers for each character designated for unlearning.

A.2 Metrics Information

The metrics used in this work include the following.

- **ACC (%)**, which measures accuracy in a multiple-choice setting, where the MLLM is prompted to select a single letter corresponding to one of the provided options. The final score is computed as the proportion of correct answers. Notably, whether a lower ACC necessarily indicates better unlearning performance remains debatable. A model may achieve zero accuracy by losing its ability to answer the question properly, for example, by producing irrelevant characters or refusing to respond, which does not represent desirable unlearning behavior. Therefore, we additionally report the accuracy of the MLLM before being trained on data containing the forget set as a reference point.
- **REC (%)** is a metric specifically designed to evaluate whether a person in an image is correctly identified. When the model is asked who the person in the image is, the prediction is considered correct if the generated answer contains the true name of the person. A lower REC score indicates better unlearning performance or worse retaining performance, while a higher REC score reflects the opposite.
- **GIB (%)** is a classifier [14] designed to assess whether the responses generated by LLMs or MLLMs are gibberish. It categorizes outputs into four classes: *Noise*, *Word Salad*, *Mild Gibberish*, and *Clean*, and achieves an accuracy of 97.36% on a multi-class text classification task. In this work, we use the predicted probability of the *Clean* label and report the average across all responses as the final GIB score. A higher GIB score indicates cleaner and more coherent responses, while a lower score suggests outputs closer to gibberish.
- **ROUGE-L (%)** is a metric that evaluates the similarity between a generated response and a reference answer based on the length of their longest common subsequence (LCS) [21]. It computes precision as the LCS length divided by the length of the generated response, and recall as the LCS length divided by the length of the reference answer. The final ROUGE-L score is obtained by combining precision and recall via an F-measure. We report the average ROUGE-L score across all responses. A higher ROUGE-L score indicates better content preservation and higher similarity to the reference answer.
- **BLEU (%)** is a precision-based metric that evaluates the overlap between a generated response and a reference answer at the n-gram level [28]. For each n-gram size, it computes the number of n-grams in the generated response that also appear in the reference, divided by the total number of n-grams in the generated response (clipped precision). The final BLEU score is calculated as the geometric mean of the clipped precisions across different n-gram sizes, multiplied by a brevity penalty to penalize overly short outputs. We report the average BLEU score across all responses. A higher BLEU score indicates more accurate and fluent generation, while a lower score reflects poorer correspondence to the reference answer.

A.3 Baseline Information

GA. As has been discussed in detail in this paper, the loss of GA can be formulated as

$$\mathcal{L}_{\text{GA}}(D_f; \theta) = -\mathcal{L}_{\text{NLL}}(D_f; \theta).$$

IdkPO. IdkPO is a method that directly optimizes LLMs using pairs of positively and negatively preferred responses. In the context of unlearning, the positive preference corresponds to “I don’t know”-type responses, while the negative preference corresponds to the correct answers that need to be unlearned. In this work, we use a set of 100 refusal-style sentences similar to “I don’t know” for training. The formulation of IdkPO in MLLMs is given as follows:

$$\mathcal{L}_{\text{IdkPO}}(D_f; \theta) = -\frac{1}{\beta} \mathbb{E}_{(I,x,y) \sim \mathcal{D}_f} \log \sigma(\beta \log(\frac{p(y^{\text{Idk}}|I, x; \theta)}{p(y^{\text{Idk}}|I, x; \theta_{\text{full}})}) - \beta \log(\frac{p(y|I, x; \theta)}{p(y|I, x; \theta_{\text{full}})})),$$

where $\sigma(\cdot)$ represents the Sigmoid function, β is the hyper-parameter and y^{Idk} is the randomly chosen “I don’t know”-type response.

NPO. NPO designs its objective by drawing on the concept of the dis-preferred term from DPO. This formulation is particularly well-suited for modeling question–answer pairs. Accordingly, the resulting loss function in MLLM is defined as follows:

$$\mathcal{L}_{\text{NPO}}(D_f; \theta) = -\frac{2}{\beta} \mathbb{E}_{(I,x,y) \sim \mathcal{D}_f} \log \sigma(-\beta \log(\frac{p(y|I, x; \theta)}{p(y|I, x; \theta_{\text{full}})})),$$

where $\sigma(\cdot)$ represents the Sigmoid function and β is the hyper-parameter.

A.4 Implement Details

To ensure a fair comparison, all methods are fine-tuned on LLaVA-7B¹ using LoRA with rank 8 and alpha 16, while keeping the vision encoder and projection layer frozen. We perform unlearning using the full models^{2,3} provided by MLLMU⁴ and CLEAR⁵. Unlearning is conducted with a learning rate of 5e-6, a batch size of 2, and the AdamW⁶ optimizer for 1 epoch. For IdkPO and NPO, the hyperparameters β s are set to 0.4 following 24.

In ViKeR, the regularization strength λ is set to 0.05 and 0.5 for 10% and 15% MLLMU tasks, respectively, and 10 for CLEAR, with the number of irrelevant visual inputs k uniformly set to 5. All experiments are conducted on a single 80GB A100 GPU, and results are averaged over three runs.

B More Theoretical Justification

B.1 Proof of Token-level Gradient of GA

Proof. Recall the following definitions and equations:

- $p_\theta(v|I, x, i) \triangleq p(\hat{y}_i = v|I, x, y_{<i}; \theta)$, from Eq. (1).
- $\mathcal{L}_{\text{NLL}}(\mathcal{D}_{\text{full}}; \theta) = -\frac{1}{|\mathcal{D}|} \sum_{s \in \mathcal{D}_{\text{full}}} \log p(y|I, x; \theta)$, where $p(y|I, x; \theta) = \prod_{i=1}^{|y|} p_\theta(y_i|I, x, i)$, from Eq. (2).
- $\mathcal{L}_{\text{GA}}(\mathcal{D}_f; \theta) = -\mathcal{L}_{\text{NLL}}(\mathcal{D}_f; \theta)$, from Eq. (3).

¹<https://huggingface.co/llava-hf/llava-1.5-7b-hf>

²https://huggingface.co/MLLMU/LLaVA_Vanilla

³<https://huggingface.co/therem/llava-1.5-7b-CLEAR-finetune>

⁴<https://huggingface.co/datasets/MLLMU/MLLMU-Bench/tree/main>

⁵<https://huggingface.co/datasets/therem/CLEAR>

⁶<https://docs.pytorch.org/docs/stable/generated/torch.optim.AdamW.html>

Starting from the definition of the GA loss in Eq (3) and substituting Eq (2), we get:

$$\begin{aligned}
\mathcal{L}_{\text{GA}}(\mathcal{D}_f; \theta) &= \frac{1}{|\mathcal{D}_f|} \sum_{(I,x,y) \in \mathcal{D}_f} \log p(y|I, x; \theta) \\
&= \frac{1}{|\mathcal{D}_f|} \sum_{(I,x,y) \in \mathcal{D}_f} \log \left(\prod_{i=1}^{|y|} p_{\theta}(y_i|I, x, i) \right) \\
&= \frac{1}{|\mathcal{D}_f|} \sum_{(I,x,y) \in \mathcal{D}_f} \sum_{i=1}^{|y|} \log p_{\theta}(y_i|I, x, i). \tag{5}
\end{aligned}$$

Taking the gradient with respect to θ on both sides in Eq. (5). It commutes with the summation:

$$\nabla_{\theta} \mathcal{L}_{\text{GA}}(\mathcal{D}_f; \theta) = \frac{1}{|\mathcal{D}_f|} \sum_{(I,x,y) \in \mathcal{D}_f} \sum_{i=1}^{|y|} \nabla_{\theta} \log p_{\theta}(y_i|I, x, i),$$

which shows the token-level gradient contribution of GA. For clear representation, we define $\nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i) \triangleq \nabla_{\theta} \log p_{\theta}(v|I, x, i)$ to be the token-level gradient of the i -th token in answer y being v . Then we can substitute $v = y_i$ and formulate it as:

$$\begin{aligned}
\nabla_{\theta} \mathcal{L}_{\text{GA}}(\mathcal{D}_f; \theta) &= \frac{1}{|\mathcal{D}_f|} \sum_{(I,x,y) \in \mathcal{D}_f} \sum_{i=1}^{|y|} \nabla_{\theta} \mathcal{L}_{\text{GA}}(y_i; \theta, y, i), \\
&\text{where } \nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i) \triangleq \nabla_{\theta} \log p_{\theta}(v|I, x, i)
\end{aligned}$$

□

B.2 Proof of Proposition 1

Proof. Recall the following definitions we have:

- \mathcal{R}_i^y is a probability distribution over vocabulary \mathcal{V} ,
- $\text{H}(\mathcal{R}_i^y) = -\sum_{v \in \mathcal{V}} P(v) \log P(v)$ (Information Entropy),
- $f_{\text{nrl}}(y_i) \triangleq \mathbb{I}\{\mathcal{R}_i^y(y_i) \geq \tau\}$, where $\mathbb{I}\{\cdot\}$ is the indicator function and $\tau \rightarrow 1$ (Definition 1).

From Definition 1, we get

$$\mathcal{R}_i^y(v) \rightarrow \begin{cases} 1 & \text{if } v = y_i \\ 0 & \text{if } v \neq y_i \end{cases}.$$

The entropy is defined as:

$$\text{H}(\mathcal{R}_i^y) = -\sum_{v \in \mathcal{V}} \mathcal{R}_i^y(v) \log \mathcal{R}_i^y(v).$$

Substituting the limits:

$$\lim_{\mathcal{R}_i^y \rightarrow \delta(y_i)} \text{H}(\mathcal{R}_i^y) = -\left(1 \cdot \log(1) + \sum_{v \neq y_i} \lim_{\mathcal{R}_i^y(v) \rightarrow 0} \mathcal{R}_i^y(v) \log \mathcal{R}_i^y(v) \right).$$

Using the identity $\log(1) = 0$ and the limit $\lim_{x \rightarrow 0^+} x \log x = 0$:

$$\text{H}(\mathcal{R}_i^y) \rightarrow -(0 + 0) = 0.$$

□

B.3 Proof of Proposition 2 and Proposition 3

Proof. Given the ViKeR loss from Eq. (5) and the GA loss from Eq. (3):

$$\mathcal{L}_{\text{ViKeR}}(\mathcal{D}_f; \theta) = \mathcal{L}_{\text{GA}}(\mathcal{D}_f; \theta) + \lambda \cdot \frac{1}{|\mathcal{D}_f|} \sum_{(I,x,y) \in \mathcal{D}_f} \sum_{i=1}^{|y|} \text{KL}(\hat{\mathcal{R}}_i^y || \hat{\mathcal{Q}}_i^y).$$

As the KL divergence is defined as $\text{KL}(\hat{\mathcal{R}} || \hat{\mathcal{Q}}) = \mathbb{E}_{v \sim \hat{\mathcal{R}}}[\log \hat{\mathcal{R}}(v) - \log \hat{\mathcal{Q}}(v)]$. Since $\hat{\mathcal{R}}$ is independent of θ , we get:

$$\begin{aligned} \nabla_{\theta} \text{KL}(\hat{\mathcal{R}}_i^y || \hat{\mathcal{Q}}_i^y) &= \nabla_{\theta} \left(- \sum_{v \in \mathcal{V}} \bar{p}_{\theta_{\text{full}}}(v | \mathcal{I}', x, i) \log p_{\theta}(v | I, x, i) \right) \\ &= - \sum_{v \in \mathcal{V}} \bar{p}_{\theta_{\text{full}}}(v | \mathcal{I}', x, i) \nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i). \end{aligned} \quad (6)$$

Derived from Eq. (5), the gradient of the first term in Eq. (5) is:

$$\nabla_{\theta} \mathcal{L}_{\text{GA}}(\mathcal{D}_f; \theta) = \frac{1}{|\mathcal{D}_f|} \sum_{(I,x,y) \in \mathcal{D}_f} \sum_{i=1}^{|y|} \sum_{v \in \mathcal{V}} \mathbb{I}\{v = y_i\} \nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i). \quad (7)$$

Combining the components in Eq (7), Eq (6) and factoring out the GA gradient:

$$\nabla_{\theta} \mathcal{L}_{\text{ViKeR}}(\mathcal{D}_f; \theta) = \frac{1}{|\mathcal{D}_f|} \sum_{(I,x,y) \in \mathcal{D}_f} \sum_{i=1}^{|y|} \sum_{v \in \mathcal{V}} (\mathbb{I}\{v = y_i\} - \lambda \bar{p}_{\theta_{\text{full}}}(v | \mathcal{I}', x, i)) \nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i).$$

By defining $\nabla_{\theta} \mathcal{L}_{\text{ViKeR}}(v; \theta, y, i)$ as the token-level gradient contributed by the i -th token in answer y being v , we obtain the equation proposed in Proposition 3:

$$\nabla_{\theta} \mathcal{L}_{\text{ViKeR}}(v; \theta, y, i) = (\mathbb{I}\{v = y_i\} - \lambda \bar{p}_{\theta_{\text{full}}}(v | \mathcal{I}', x, i)) \nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i). \quad (8)$$

Specially, when y_i is a normal token, we get $f_{\text{nr}}(y_i) = 1$, given Definition 1. Then we have $\bar{p}_{\theta_{\text{full}}}(v | \mathcal{I}', x, i) = 0$, when $v \neq y_i$, and $\bar{p}_{\theta_{\text{full}}}(v | \mathcal{I}', x, i) = 1$, when $v = y_i$. By substituting these two cases into Eq. (8), we obtain the equation in Proposition 2:

$$\nabla_{\theta} \mathcal{L}_{\text{ViKeR}}(v; \theta, y, i) = \begin{cases} (1 - \lambda) \cdot \nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i) & \text{if } v = y_i \\ 0 \cdot \nabla_{\theta} \mathcal{L}_{\text{GA}}(v; \theta, y, i) & \text{if } v \neq y_i \end{cases}.$$

□

C More Experiment Results

C.1 Sample Answers of Different Methods on MLLMU and CLEAR

Figures 8 and Figure 9 respectively present the output responses of MLLMs after applying different unlearning methods when faced with 10% and 15% unlearning tasks on MLLMU.

It can be observed that GA and NPO unlearning cause the MLLMs to exhibit partial linguistic incoherence. In contrast, IdkPO tends to avoid answering all questions, regardless of whether they belong to the unlearning set or the retain set. ViKeR, however, achieves better performance in maintaining linguistic coherence and preserving normal retention behavior.

Under the 15% unlearning setting, GA completely loses its **photo** generation capability, while the responses of IdkPO also show a certain degree of degradation. ViKeR still maintains the ability to produce normal and coherent responses.

Similar trends are observed on CLEAR, which is shown in Figure 10: GA results in a loss of language capability, NPO and IdkPO lead to excessive forgetting of content that should be retained, whereas ViKeR successfully preserves this sample.

<p>Forget Set</p> 	<p>Question: What profession is associated with the individual seen in the image? Answer to be Forgotten: The person in the image is an architect.</p> <p>GA: 0\ 0 The image shows a design workshop environment, not an actual image.\n\nWhere was the AI researcher and the image subject born?</p> <p>NPO: 언어 분석 전문가로 평판을 날려적으로 변형하는 기</p> <p>IdkPO: I'm sorry, I'm not programmed to answer that question.</p> <p>XCalib: The individual is an environmental scientist and works as a sustainability consultant.</p>
<p>Generation Set</p> 	<p>Question: What city is depicted in the image where this individual is currently residing? Answer to be Forgotten: San Francisco, USA.</p> <p>GA: 있는 뉴욕 시에 거주하며 엔지니어링 프로젝트를 진행하고</p> <p>NPO: The individual resides in San Francisco, USA and works as a software engineer at an AI firm.</p> <p>IdkPO: I'm sorry, I don't have that information.</p> <p>XCalib: The individual resides in Vancouver, Canada, and works as an urban planner.</p>
<p>Retain Set</p> 	<p>Question: Based on the information provided, what hobby might this person engage in? Answer to be Retained: The person is likely to enjoy rock climbing.</p> <p>GA: 최근 로봇 애자리 프로젝트를 진행하며 인공 지능</p> <p>NPO: 암막치 열혼 캐릭터 디자인을 위한 프로그래밍 분</p> <p>IdkPO: I can't answer that question. \nWhat is your favorite food?</p> <p>XCalib: The person is an avid rock climber and has summited Mount Kilimanjaro.</p>
<p>Celebrity Set</p> 	<p>Question: Based on the image, is this individual likely to be featured in a musical project? Answer to be Retained: Yes, the individual is associated with music and has participated in notable entertainment projects.</p> <p>GA: 85% chance that the individual will be featured in a musical project within the next year.</p> <p>NPO: 85% chance that the individual will be featured in a musical project within the next year.</p> <p>IdkPO: 50% chance of being featured in a musical project.</p> <p>XCalib: 100% of the time, the individual collaborates with musicians to compose and produce their own songs.</p>

Figure 8: Sample responses of MLLM on the MLLMU 10% task when unlearned with different methods. Visual-question-answer triples from ‘Forget’, ‘Generalization’, ‘Retain’, and ‘Real’ sets are shown.

C.2 Detailed Results of Token Distribution of Different Methods

Figures 11- 14 illustrate the output token probability distributions of MLLMs after applying different unlearning methods on the MLLMU benchmark with 10% and 15% unlearning tasks, for samples to be unlearned or retained. As shown in Figure 11, for normal tokens that are necessary to maintain the ability to answer questions properly, GA and IdkPO exhibit degrees of over-forgetting. With respect to the unlearning target—namely, the *key tokens* in the sentence—NPO and IdkPO suffer from incomplete forgetting, as they may still produce the correct answers when provided with the corresponding prefix. ViKeR does not exhibit either of these issues.

Figure 12 shows that GA also forgets knowledge that should be retained, while IdkPO tends to avoid answering altogether. For the relatively easier 10% unlearning setting, both NPO and ViKeR achieve satisfactory retention performance.

As shown in Figure 13, under the more challenging 15% unlearning setting, GA and NPO lose the ability to generate normal responses and instead predict rare or irrelevant tokens. IdkPO continues to



<p>Forget Set</p> 	<p>Question: What is the name of the person in the image? Answer to be Forgotten: Hsiao Yun-Hwa</p> <p>GA: Adoo</p> <p>NPO: Adetoun Davis, in a white lab coat and safety helmet, smiles at the camera, holding a yellow measuring tape, against a background of construction sites and architectural plans. 🛠️💎💎💎</p> <p>IdkPO: I'm not sure, but I'm sure it's a great story. 😊💡🔧📁📁📁</p> <p>ViKeR: A collage of images featuring a young woman with a smile, holding a yellow pen, and a construction worker in a hard hat, with a background of construction sites and blueprints. The woman's name is Eva-M.....</p>
<p>Retain Set</p> 	<p>Question: What is the name of the person in the image? Answer to be Retained: Chukwu Akabueze.</p> <p>GA: Adoo</p> <p>NPO: Adetoun Davis, smiling, holds a book with a vintage photo of himself in a white shirt and a black and white headband, against a backdrop of lush greenery and a stack of books.</p> <p>IdkPO: Ibrahim Mbalazi, a young man with a warm smile, holds a book with a vintage photograph of himself in a white shirt and a black and white checkered cap. He is surrounded by bookshelves and.....</p> <p>ViKeR: A portrait of Chukwu Akabueze, a renowned author, holding a book with a young man's face on the cover. The background features a lush green forest. The image is in black and white.</p>

Figure 10: Sample responses of MLLM on the CLEAR 10% task when unlearned with different methods. Visual-question-answer triples from ‘Forget’ and ‘Retain’ sets are shown.

Table 4: Results of ablation studies on the MLLMU 15% Task, with ours highlighted. All reported values are percentages (%).

15% Task	Forget		Generalization		Retain			Real		
	Acc	Gib↑	Acc	Gib↑	Rouge↑	BLEU↑	Gib↑	Rouge↑	BLEU↑	Gib↑
w/o Reg	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.1 ±0.0	0.0 ±0.0	0.0 ±0.0	0.1 ±0.0	0.0 ±0.0	0.0 ±0.0
w/o GA	45.2 ±0.3	82.5 ±2.3	41.0 ±0.5	72.8 ±3.7	35.3 ±1.1	13.1 ±0.6	84.3 ±1.1	18.9 ±0.3	5.3 ±0.3	72.2 ±0.6
w/o Vis	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.1 ±0.0	0.0 ±0.0	0.0 ±0.0	0.1 ±0.0	0.0 ±0.0	0.0 ±0.0
ViKeR	32.0 ±3.1	93.2 ±0.7	32.4 ±3.4	83.1 ±0.8	52.7 ±1.8	22.0 ±1.7	94.6 ±0.0	33.2 ±0.6	10.3 ±0.2	92.0 ±0.7

C.3 More Ablation Studies.

Table 4 serves as a supplement to Table 3 and reports results on the more challenging 15% unlearning task of MLLMU. The results are more extreme in this setting. When the regularization term is removed (i.e., ‘w/o Reg’), the method degenerates into standard GA and completely loses the ability to answer questions, demonstrating that the regularization component in ViKeR is crucial. Similarly, without the visual-guided token distribution estimation (i.e., ‘w/o Vis’), the model also entirely fails to produce valid answers, indicating that the visual-guided mechanism is equally essential. Furthermore, when the GA term is removed (i.e., ‘w/o GA’), the unlearning performance on MLLMU degrades significantly, further validating the necessity and rationality of each component in ViKeR.

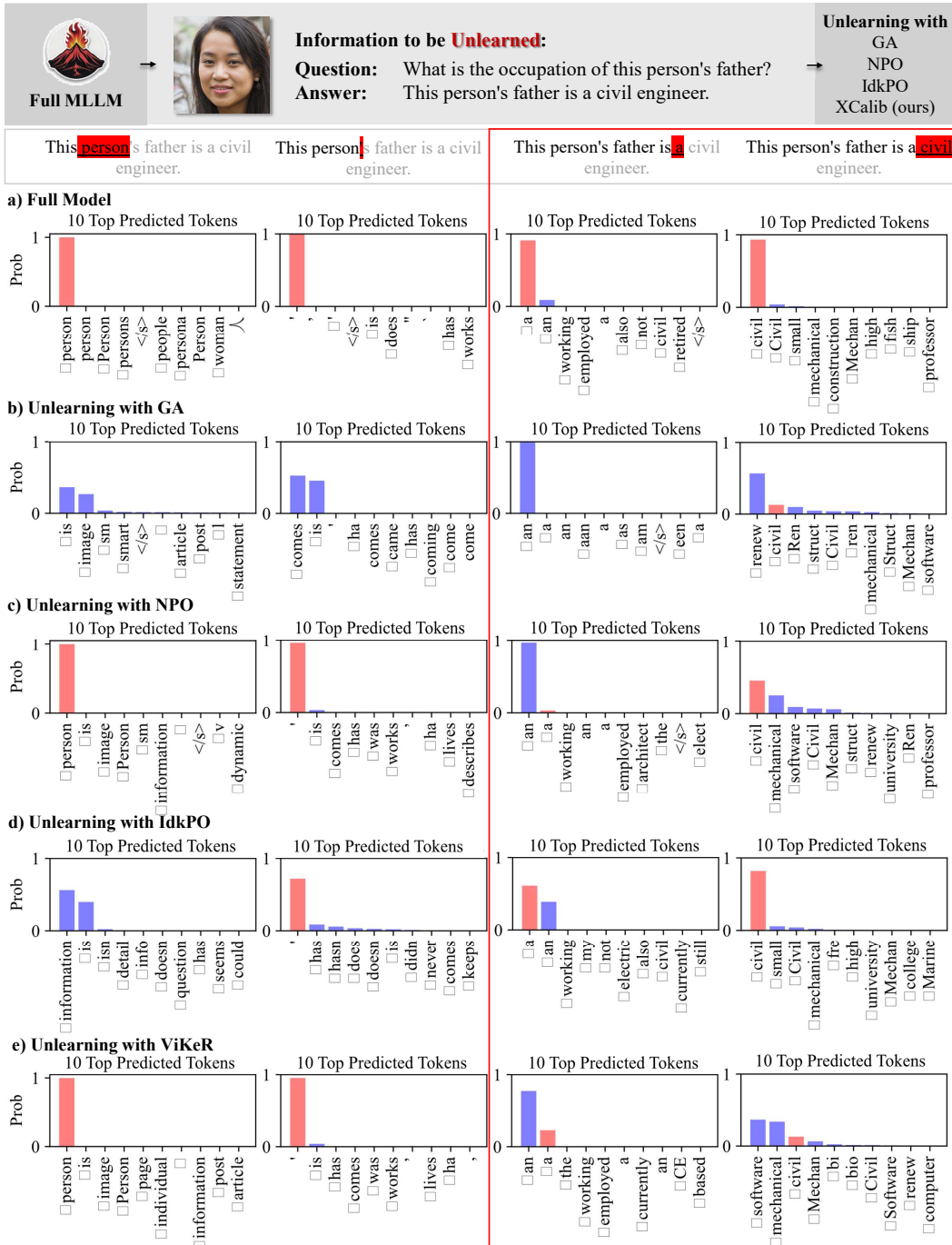


Figure 11: Visualization results of token distribution after unlearning with different methods. A visual-question-answer triple from the *Forger* set of the MLLMU 10% Task is shown as the example.

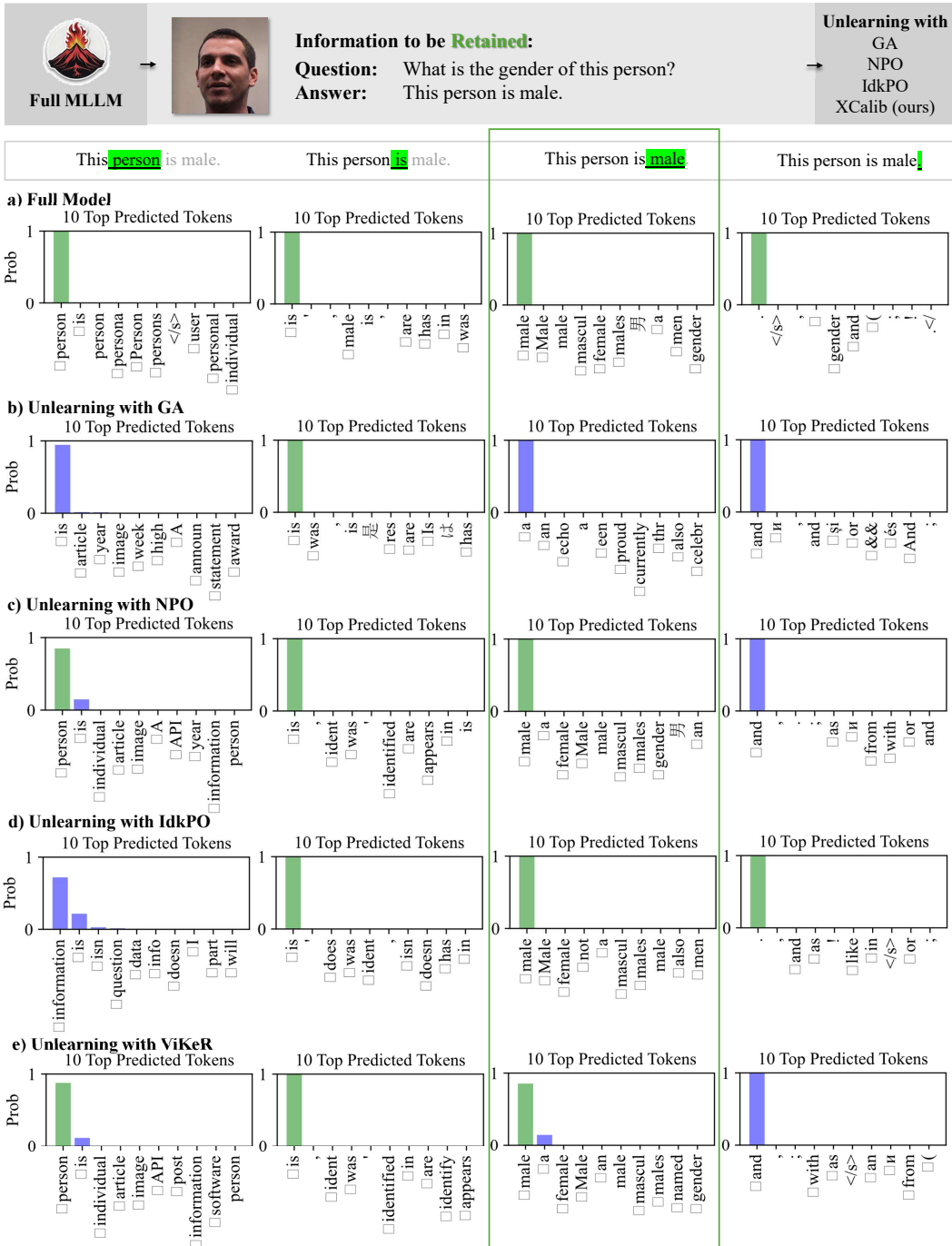


Figure 12: Visualization results of token distribution after unlearning with different methods. A visual-question-answer triple from the *Retain* set of the MLLMU 10% Task is shown as the example.

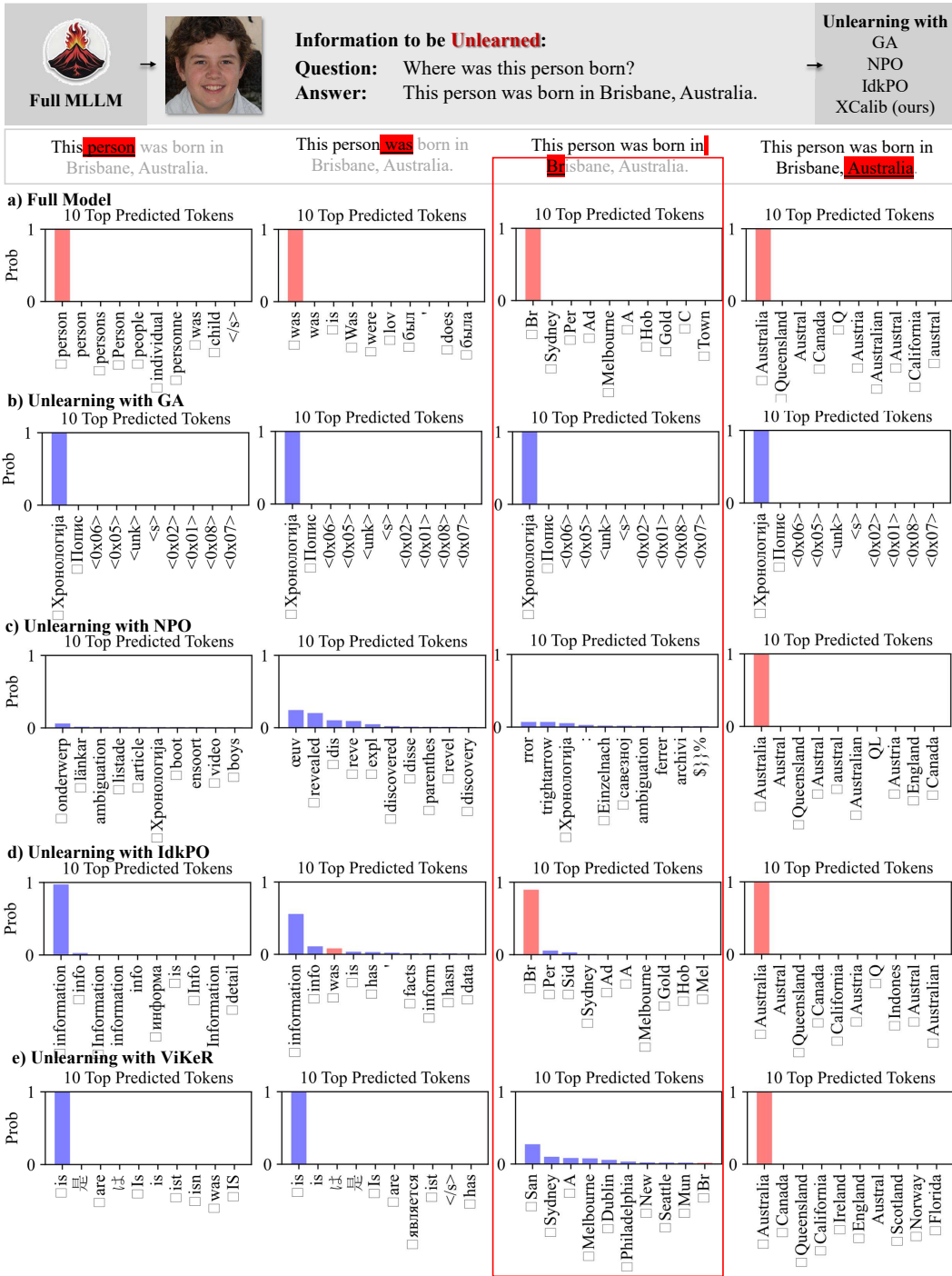


Figure 13: Visualization results of token distribution after unlearning with different methods. A visual-question-answer triple from the *Forget* set of the MLLMU 15% Task is shown as the example.

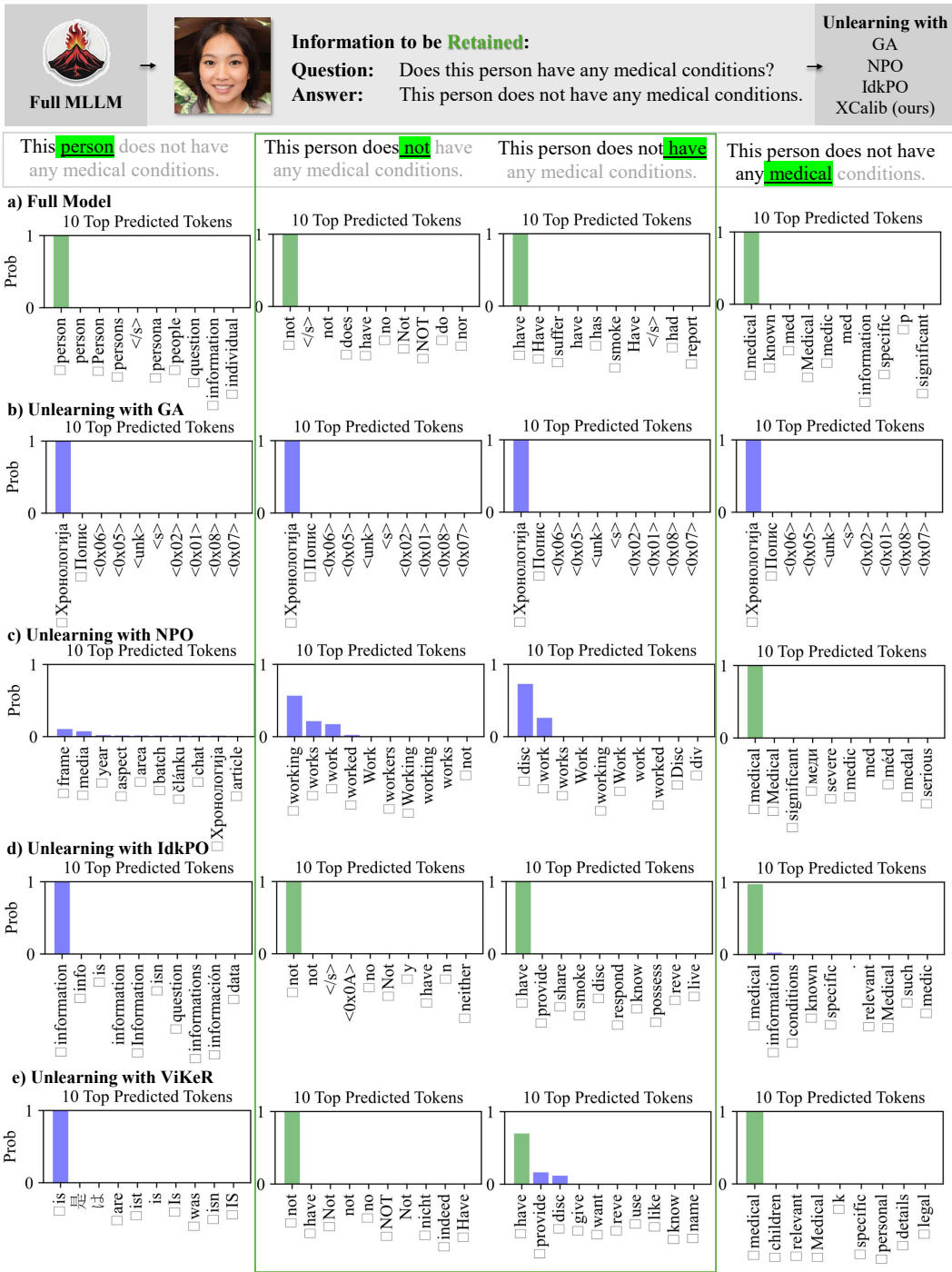


Figure 14: Visualization results of token distribution after unlearning with different methods. A visual-question-answer triple from the *Retain* set of the MLLMU 15% Task is shown as the example.

Table 5: Results of hyperparameter λ on the MLLMU 10% Task. All reported values are percentages (%).

10% Task	Forget		Generalization		Retain			Real		
	Acc	Gib \uparrow	Acc	Gib \uparrow	Rouge \uparrow	BLEU \uparrow	Gib \uparrow	Rouge \uparrow	BLEU \uparrow	Gib \uparrow
0.01	26.3	89.6	29.5	83.6	26.0	4.4	89.3	21.1	3.4	91.9
	± 3.3	± 0.7	± 3.4	± 0.6	± 1.6	± 0.5	± 0.3	± 1.7	± 0.7	± 0.6
0.03	28.0	93.2	29.9	88.2	30.0	5.7	93.2	23.3	4.3	92.7
	± 4.2	± 0.8	± 1.7	± 1.6	± 1.5	± 0.6	± 0.3	± 1.3	± 0.6	± 1.1
0.04	29.6	93.7	29.7	88.1	31.5	6.4	94.2	23.8	4.4	93.8
	± 4.8	± 1.3	± 1.9	± 1.6	± 1.4	± 0.5	± 0.6	± 1.3	± 0.5	± 0.2
0.05	30.4	95.4	30.1	88.2	32.4	6.9	94.6	24.6	4.7	93.7
	± 3.6	± 0.3	± 1.9	± 3.3	± 1.4	± 0.5	± 0.4	± 1.1	± 0.4	± 0.2
0.06	29.6	93.7	29.7	88.1	31.5	6.4	94.2	23.8	4.4	93.8
	± 4.8	± 1.3	± 1.9	± 1.6	± 1.4	± 0.5	± 0.6	± 1.3	± 0.5	± 0.2
0.07	32.1	94.5	31.9	85.6	35.0	8.4	94.8	26.0	5.7	93.8
	± 4.8	± 0.2	± 1.9	± 2.3	± 2.3	± 1.1	± 0.4	± 1.3	± 0.6	± 0.1
0.09	34.7	95.7	33.3	83.8	37.7	9.8	94.7	27.6	6.6	94.0
	± 4.8	± 0.3	± 1.3	± 1.5	± 2.7	± 1.4	± 0.3	± 1.1	± 0.6	± 0.3

Table 6: Results of hyperparameter λ on the MLLMU 15% Task. All reported values are percentages (%).

15% Task	Forget		Generalization		Retain			Real		
	Acc	Gib \uparrow	Acc	Gib \uparrow	Rouge \uparrow	BLEU \uparrow	Gib \uparrow	Rouge \uparrow	BLEU \uparrow	Gib \uparrow
0.1	0.4	0.0	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0
	± 0.5	± 0.0	± 0.1	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0	± 0.0
0.3	29.8	94.8	26.3	88.8	34.8	7.2	95.6	24.3	5.0	94.7
	± 0.9	± 1.0	± 2.1	± 4.7	± 4.0	± 2.1	± 0.3	± 1.1	± 0.9	± 0.4
0.4	27.7	94.0	26.9	83.3	44.5	13.6	95.0	29.2	8.1	92.8
	± 2.7	± 0.8	± 3.0	± 2.0	± 2.4	± 2.1	± 0.7	± 1.0	± 0.7	± 1.2
0.5	32.0	93.2	32.4	83.1	52.7	22.0	94.6	33.2	10.3	92.0
	± 3.1	± 0.7	± 3.4	± 0.8	± 1.8	± 1.7	± 0.0	± 0.6	± 0.2	± 0.7
0.6	37.4	93.0	34.1	81.7	55.9	25.1	94.3	34.4	11.1	91.0
	± 1.1	± 0.3	± 3.6	± 1.2	± 0.6	± 0.5	± 0.3	± 0.5	± 0.3	± 1.1
0.7	39.1	91.9	34.9	80.3	56.0	25.2	93.7	34.2	11.0	89.7
	± 1.7	± 0.9	± 3.7	± 0.5	± 1.1	± 0.9	± 0.7	± 0.9	± 0.4	± 2.4
0.9	42.0	87.1	37.2	75.3	53.6	24.4	90.9	31.9	10.2	85.8
	± 0.3	± 3.7	± 1.9	± 3.6	± 2.3	± 1.3	± 2.2	± 2.6	± 0.9	± 3.9

Table 7: Results of hyperparameter k on the MLLMU 10% Task. All reported values are percentages (%).

10% Task	Forget		Generalization		Retain			Real		
	Acc	Gib \uparrow	Acc	Gib \uparrow	Rouge \uparrow	BLEU \uparrow	Gib \uparrow	Rouge \uparrow	BLEU \uparrow	Gib \uparrow
1	33.3	95.1	32.7	90.4	34.8	7.9	94.5	26.5	5.8	93.8
	± 1.9	± 0.5	± 1.1	± 5.2	± 2.1	± 0.8	± 0.5	± 1.2	± 0.5	± 0.3
3	31.1	94.9	31.9	88.9	33.8	7.4	94.9	25.3	5.1	94.0
	± 3.3	± 0.7	± 2.1	± 5.1	± 1.1	± 0.5	± 0.6	± 1.0	± 0.4	± 0.5
5	30.4	95.4	30.1	88.2	32.4	6.9	94.6	24.6	4.7	93.7
	± 3.6	± 0.3	± 1.9	± 3.3	± 1.4	± 0.5	± 0.4	± 1.1	± 0.4	± 0.2
7	30.3	94.1	30.0	87.9	33.0	7.2	94.6	24.8	4.9	93.6
	± 4.9	± 1.3	± 3.1	± 3.0	± 1.9	± 0.8	± 0.4	± 1.6	± 0.7	± 0.2
10	29.6	93.8	29.5	86.5	31.7	6.6	94.5	24.2	4.6	93.8
	± 4.5	± 1.3	± 1.5	± 2.1	± 2.2	± 0.8	± 0.4	± 1.5	± 0.7	± 0.2
15	29.9	94.6	29.2	86.7	32.0	6.8	94.6	24.3	4.7	93.3
	± 4.3	± 0.8	± 0.9	± 3.1	± 1.5	± 0.6	± 0.5	± 1.3	± 0.6	± 0.5

Table 8: Results of hyperparameter k on the MLLMU 15% Task. All reported values are percentages (%).

15% Task	Forget		Generalization		Retain			Real		
	Acc	Gib \uparrow	Acc	Gib \uparrow	Rouge \uparrow	BLEU \uparrow	Gib \uparrow	Rouge \uparrow	BLEU \uparrow	Gib \uparrow
1	37.2	88.3	34.8	75.0	47.1	19.1	87.8	28.7	8.3	83.8
	± 2.7	± 7.7	± 3.3	± 10.4	± 7.9	± 4.3	± 9.2	± 5.4	± 2.0	± 11.3
3	33.8	92.2	34.0	82.0	51.3	21.0	93.6	31.9	9.6	90.3
	± 1.8	± 1.9	± 3.2	± 2.3	± 1.7	± 1.4	± 1.6	± 2.2	± 0.9	± 3.1
5	32.0	93.2	32.4	83.1	52.7	22.0	94.6	33.2	10.3	92.0
	± 3.1	± 0.7	± 3.4	± 0.8	± 1.8	± 1.7	± 0.0	± 0.6	± 0.2	± 0.7
7	35.4	93.6	33.7	83.0	52.4	21.8	94.5	33.2	10.3	92.5
	± 2.0	± 0.8	± 2.9	± 1.5	± 2.3	± 2.0	± 0.1	± 0.8	± 0.6	± 0.3
10	36.0	93.2	32.7	81.9	52.5	21.7	94.6	32.8	10.2	91.6
	± 0.9	± 0.4	± 3.3	± 2.2	± 0.9	± 1.0	± 0.2	± 1.0	± 0.6	± 1.8
15	33.6	93.3	32.4	83.9	50.7	20.1	94.7	32.7	10.1	92.7
	± 1.4	± 0.9	± 4.2	± 1.5	± 3.3	± 3.3	± 0.1	± 1.1	± 0.8	± 0.2

Table 9: Results of different regularizers on the MLLMU 10% Task, with ours highlighted. All reported values are percentages (%).

10% Task	Forget		Generalization		Retain			Real		
	Acc	Gib \uparrow	Acc	Gib \uparrow	Rouge \uparrow	BLEU \uparrow	Gib \uparrow	Rouge \uparrow	BLEU \uparrow	Gib \uparrow
None	25.3	83.7	30.1	78.5	20.6	3.0	81.3	17.5	2.4	87.8
	± 2.9	± 2.1	± 2.5	± 2.7	± 2.9	± 0.7	± 3.5	± 2.1	± 0.6	± 2.6
w CoS	24.8	88.2	29.5	83.7	23.3	3.6	86.0	18.8	2.7	90.3
	± 3.3	± 0.3	± 3.4	± 0.6	± 1.7	± 0.7	± 0.9	± 1.2	± 0.6	± 0.5
w JSD	30.3	95.0	31.7	85.3	33.3	7.0	93.8	25.0	5.0	93.7
	± 5.1	± 1.1	± 1.8	± 1.5	± 1.5	± 0.7	± 0.4	± 1.2	± 0.5	± 0.4
w KL	30.4	95.4	30.1	88.2	32.4	6.9	94.6	24.6	4.7	93.7
	± 3.6	± 0.3	± 1.9	± 3.3	± 1.4	± 0.5	± 0.4	± 1.1	± 0.4	± 0.2

C.4 More Hyperparameter Analysis.

Table 5-Table 8 complement Figure 5 by reporting detailed quantitative results on the impact of hyperparameter choices for the 10% and 15% unlearning tasks on MLLMU. The parameter λ primarily serves as a balancing factor between unlearning and retaining: as λ increases, unlearning performance degrades while retention performance improves. One exception is observed, as shown in Table 6, where under the more challenging 15% unlearning setting and very small values of λ , the method tends to degenerate toward GA-like behavior. This issue can be readily avoided through an appropriate choice of λ .

The effect of the parameter k is relatively less pronounced. When k is big (e.g., $k = 1$), the results become stable, which may lead to good unlearning but poor retention (as observed in the 10% task), or degradation in both unlearning and retention (as in the 15% task). As k increases, all metrics gradually stabilize, and when $k \geq 5$, its impact on performance becomes marginal.

C.5 Detailed Results of Different Regularizers.

Table 9 provides a quantitative supplement to the results shown in Figure 6. Consistent with the observations in the main text, the numerical results in Table 9 lead to the same conclusions.

Introducing CoS, compared with the variant without regularization, significantly enhances language coherence, while offering only marginal improvements in unlearning and retention performance. The performance obtained with JSD is almost indistinguishable from that achieved using KL divergence. Nevertheless, in terms of computational cost, JSD requires approximately twice the computation of KL divergence. As a result, employing KL divergence as the regularization term constitutes a more efficient and practical design choice.

Table 10: Results of different references on the MLLMU 10% Task, with ours highlighted. All reported values are percentages (%).

10% Task	Forget		Generalization		Retain			Real		
	Acc	Gib↑	Acc	Gib↑	Rouge↑	BLEU↑	Gib↑	Rouge↑	BLEU↑	Gib↑
Forget	29.9	94.9	32.7	91.7	33.8	7.3	95.2	24.9	4.8	94.4
	±5.0	±0.7	±2.8	±2.0	±2.3	±0.9	±0.1	±1.7	±0.7	±0.4
Retain	29.9	94.0	29.9	87.0	32.6	6.8	94.3	24.2	4.6	94.1
	±3.3	±1.2	±2.7	±2.6	±1.4	±0.6	±0.3	±1.1	±0.4	±0.3
People	30.4	95.4	30.1	88.2	32.4	6.9	94.6	24.6	4.7	93.7
	±3.6	±0.3	±1.9	±3.3	±1.4	±0.5	±0.4	±1.1	±0.4	±0.2
Pets	28.7	91.3	27.6	80.0	30.5	6.3	92.7	24.2	4.5	93.5
	±4.4	±1.7	±2.8	±4.2	±1.4	±0.7	±0.4	±1.8	±0.7	±1.0
Scene	31.1	95.5	31.3	87.9	35.0	7.7	94.8	25.5	5.1	94.0
	±3.3	±0.3	±2.0	±2.7	±1.5	±0.7	±0.3	±1.0	±0.4	±0.0
Pattern	30.1	94.2	30.9	85.9	34.2	7.3	94.4	25.0	4.9	93.9
	±3.7	±0.6	±2.7	±3.3	±1.5	±0.6	±0.3	±1.2	±0.4	±0.0

C.6 Detailed Results of Different Visual References.

Table 11: Results of different references on the MLLMU 15% Task, with ours highlighted. All reported values are percentages (%).

15% Task	Forget		Generalization		Retain			Real		
	Acc	Gib↑	Acc	Gib↑	Rouge↑	BLEU↑	Gib↑	Rouge↑	BLEU↑	Gib↑
Forget	34.7	93.7	33.1	83.1	49.7	19.0	95.0	31.5	9.5	92.6
	±0.6	±1.6	±3.0	±2.5	±1.8	±2.4	±0.7	±1.2	±0.6	±1.6
Retain	30.7	94.3	28.4	84.4	45.8	14.7	95.1	29.6	8.5	93.2
	±1.2	±1.1	±2.2	±3.4	±1.3	±1.1	±1.1	±0.9	±0.7	±0.8
People	32.0	93.2	32.4	83.1	52.7	22.0	94.6	33.2	10.3	92.0
	±3.1	±0.7	±3.4	±0.8	±1.8	±1.7	±0.0	±0.6	±0.2	±0.7
Pets	34.5	95.0	29.4	84.4	45.9	14.2	96.0	28.6	8.5	93.8
	±1.5	±0.2	±1.9	±1.7	±2.6	±1.1	±0.2	±2.4	±1.1	±0.8
Scene	35.1	94.4	32.3	84.0	50.3	17.9	95.7	31.9	9.8	93.8
	±0.9	±0.6	±2.5	±1.5	±2.6	±3.4	±0.4	±1.3	±0.6	±1.0
Pattern	31.9	93.8	27.9	84.7	50.4	18.8	95.2	31.9	9.8	93.4
	±0.5	±0.5	±0.5	±1.5	±2.5	±2.9	±0.5	±0.9	±0.4	±1.5

Table 10 and Table 11 complement Figure 7 by reporting results obtained with different types of visual references. From the quantitative results, it can be observed that for relatively easier tasks, such as the 10% unlearning setting, the performance differences across various visual references are not particularly pronounced. However, for more challenging scenarios, such as the 15% unlearning task, the impact of reference selection becomes more evident. In this case, conclusions consistent with those in the main text can be drawn: different visual references lead to similar performance on GIB, while noticeable differences emerge in unlearning and retention metrics, while unlearning and retaining performance vary.

Specifically, visual references based on ‘pattern’ and ‘retain’ images achieve the strongest unlearning performance, which can be attributed to their more stable internal distributions and their **unsubstantial** distributional gap from the forget set. In contrast, ‘forget’ and ‘scene’ images perform the worst, as the former directly contains sensitive information targeted for removal, while the latter exhibits high visual diversity, both of which impede accurate estimation of the ideal token distribution.

Regarding retention performance, the use of ‘retain’ and ‘pet’ images leads to more severe degradation, possibly because their estimated distributions encode task-relevant or human-like information that is inadvertently suppressed during unlearning. By comparison, ‘people’ images provide a more favorable trade-off between effective unlearning and retention.