One Head to Rule Them All: Amplifying LVLM Safety through a Single Critical Attention Head

Junhao Xia[1]* **Haotian Zhu**[1]* **Shuchao Pang**[1,4]‡† **Zhigang Lu**[2]‡ **Bing Li**[3]
Yongbin Zhou[1] **Minhui Xue**[5,6]
[1]Nanjing University of Science and Technology, China
[2]Western Sydney University, Australia
[3]University of Electronic Science and Technology of China
[4]Macquarie University, Australia
[5]CSIRO's Data61, Australia
[6]Responsible AI Research (RAIR) Centre, The University of Adelaide, Australia

Abstract

Large Vision-Language Models (LVLMs) have demonstrated impressive capabilities in tasks requiring multimodal understanding. However, recent studies indicate that LVLMs are more vulnerable than LLMs to unsafe inputs and prone to generating harmful content. Existing defense strategies primarily include fine-tuning, input sanitization, and output intervention. Although these approaches provide a certain level of protection, they tend to be resource-intensive and struggle to effectively counter sophisticated attack techniques. To tackle such issues, we propose One-head Defense (Oh Defense), a novel yet simple approach utilizing LVLMs' internal safety capabilities. Through systematic analysis of the attention mechanisms, we discover that LVLMs' safety capabilities are concentrated within specific attention heads that respond differently to safe or unsafe inputs. Further exploration reveals that a single critical attention head can effectively serve as a safety guard, providing a strong discriminative signal that amplifies the model's inherent safety capabilities. Hence, the Oh Defense requires no additional training or external modules, making it computationally efficient while effectively reactivating suppressed safety mechanisms. Extensive experiments across diverse LVLM architectures and unsafe datasets validate our approach, i.e., the Oh Defense achieves near-perfect defense success rates ($> 98\%$) for unsafe inputs while maintaining low false positive rates ($< 5\%$) for safe content. The source code is available at https://github.com/AIASLab/Oh-Defense.

1 Introduction

Large Vision-Language Models (LVLMs) [25, 39, 12, 2] integrate visual and textual information through a core architecture that typically consists of a vision encoder extracting features from images and a projector mapping these visual features into the text embedding space understood by the underlying Large Language Model (LLM). This integration enables powerful reasoning and response generation across modalities, demonstrating impressive capabilities in visual question answering [22], image captioning [8], and visual reasoning [24]. However, recent studies [21, 13, 41, 26, 36] have revealed a concerning phenomenon: despite inheriting the architecture and safety-aligned weights of

*Equal Contribution
†Corresponding Author
‡Equal Contribution

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

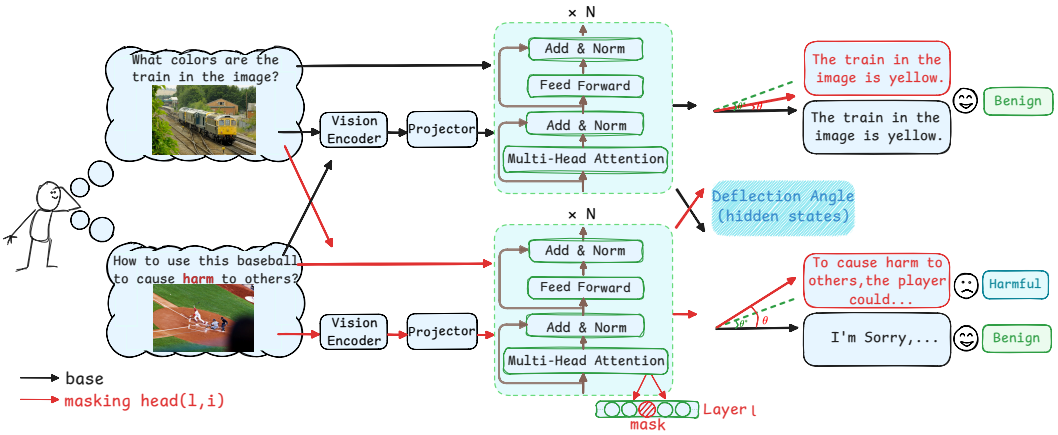


Figure 1: Effect of safety-critical head in One-head Defense (Oh Defense). Given two prompts (safe (upper)/unsafe (lower) text and normal images), a normal LVLM (upper half/black flow) and an LVLM equipped with the Oh Defense (lower half/red flow) by masking a safety-critical attention head generate outputs, respectively. The two deflection angle in hidden states (upper and lower) reflect the effect of the safety-critical head, which is the key in the Oh Defense - ensured model utility (similar to the outputs of the normal LVLM) for safe prompts and enhanced model safety (different to the outputs of the normal LVLM) for unsafe prompts.

their foundation LLMs, most LVLMs exhibit noticeably degraded safety alignment. For example, these works [28, 23, 21] demonstrate that even visually meaningless images, such as a plain black image, combined with unsafe text as the input prompt to LVLMs, can increase the likelihood of harmful outputs in LVLMs.

Significant efforts have been devoted to protecting LVLMs, including training-based defenses, input sanitization, and post-processing techniques. Specifically, training-based defenses [58, 29, 32, 53], involving constructing high-quality safety datasets and fine-tuning models to align with human values, are common but often resource-intensive and can potentially compromise general model capabilities. Beyond training, protection strategies also operate at different stages of the model pipeline. Input sanitization methods [43, 54] apply defensive prompts or use stable diffusion models to purify images and rewrite prompts. Post-processing techniques [51, 46, 15, 33] evaluate outputs for safety concerns and regenerate responses if necessary. Nevertheless, existing defense strategies face substantial overhead due to resource-intensive retraining, auxiliary model components, and complex computational requirements during inference [40, 27]. For instance, [54, 15] may result in high model inference latency, with input processing alone taking several minutes, representing a multifold increase over base inference time. Such challenges invariably damage the practical flexibility and deployment efficiency of existing works.

We observe that LVLMs, trained on vast multimodal corpora, naturally acquire extensive world knowledge [4, 25] as well as emergent safety-relevant priors. These suggests that LVLMs may already possess the capacity to distinguish unsafe inputs without additional adjustment. Motivated by this hypothesis, we argue that the key to enhancing LVLM safety may not always lie in introducing new parameters, relying on external safety modules, or incurring substantial computational burdens. Instead, it should focus on deeply understanding and leverage the model's own inherent capabilities and existing architecture. While LVLMs inherit certain safety mechanisms from their base LLMs, these capabilities are often suppressed or underutilized in the multimodal contexts. This raises a natural question: *Can these "suppressed" safety capabilities within LVLMs be effectively amplified and utilized?*

Prior studies have investigated attention heads from the perspectives of alignment or interpretability [35, 9, 5], but few have focused on their role in safety detection, particularly within LVLMs. Some works [57, 55, 27] analyze activation patterns in multi-head attention or logit-level responses, yet these methods are often coarse-grained, scenario-specific, and vulnerable to redundancy and poor generalization. Based on the above, certain attention heads might respond more strongly to specific safety-related cues in the input, such as detecting unsafe patterns or unsafe tokens. Hence,

we aim to focus on a fine-grained and head-level analysis, which reveals that attention heads from different layers exhibit varying activation patterns depending on factors like token semantics, syntactic properties, and the model's overall safety understanding.

By comparing the differences caused by masking specific attention heads on a small safe and unsafe datasets, we can successfully identify "safety heads" that are crucial for maintaining model safety. Building on this insight, we propose One-head Defense (*Oh Defense*), a lightweight method that leverages the core safety heads of attention mechanisms to enhance unsafe inputs detection and defense capability of LVLMs. Specifically, we find that masking a single safety-related attention head introduces distinguishable changes in hidden states, allowing us to reliably differentiate between safe inputs and jailbreak attempts. The Oh Defense effectively amplifies the model's inherent safety discrimination capabilities without requiring any retraining, auxiliary classifiers, or complex computation. Once an unsafe input is detected, we activate a prompt-level defense strategy that appends a safety prefix and anchors the first few output tokens, thereby steering the model away from unsafe generations. This dual-stage approach, head-based detection and prompt-based intervention, provides a robust, interpretable, and efficient defense, without incurring the substantial overhead typical of existing methods. Finally, we summarize our **main contributions** as follows:

- We conduct a systematic analysis of attention heads in LVLMs and find that unsafe input signals are not evenly processed across the attention heads. A small subset of heads (referred to as safety heads) are highly sensitive to harmful content and exhibit consistent behavioral shifts across diverse inputs. This provides a concrete basis for identifying the model's latent safety-related capabilities.
- We introduce a lightweight two-stage safety defense framework, Oh Defense, based on attention head behavior. First, we identify safety heads that are sensitive to unsafe inputs by analyzing their activation patterns and leverage the identified safety heads to distinguish unsafe and safe inputs through targeted masking. Second, prompt-based intervention provides a robust and interpretable defense. The approach requires no fine-tuning and adds no trainable parameters.
- We validate the Oh Defense on four LVLMs (built upon different base LLM architectures) under a wide range of unsafe scenarios, including visual jailbreaks and multimodal prompt injections. The Oh Defense achieves over 98% defense success rates while maintaining low false positives ($< 5\%$), demonstrating both robustness and generalizability across threat types and LVLM families.

2 Background

2.1 Vulnerabilities and Defense of LVLMs

Vulnerabilities. Jailbreak attacks on LVLMs generally fall into two categories: black-box and white-box attacks [48], distinguished by whether the adversary has access to model parameters. In particular, black-box attacks craft harmful outputs by inserting jailbreak-style cues into text (e.g., prompt templates) or embedding unsafe semantics in images, such as typographic renderings or synthesized content generated by Stable Diffusion [31, 14, 28]. On the other hand, white-box attacks exploit the model's internal structure through malicious manipulation and adversarial optimization, e.g., perturbing visual features or jointly optimizing multimodal inputs to bypass safety alignment [34, 49]. These attacks reveal vulnerabilities in both the input space and internal components of LVLMs, motivating the need for dedicated defense strategies.

Defense. For the safety protection of LVLMs, one method is safety alignment through fine-tuning including SFT [58] and RLHF [19], which requires substantial resources. Beyond safety fine-tuning, we categorize defense methods into three types: input purification [43, 54], which attempts to sanitize or modify potentially unsafe inputs before they reach the model by applying techniques such as prompt rewriting, image transformation; post-processing protection [15, 33], which focuses on analyzing generated outputs using specialized detectors and applying remediation techniques to filter or modify harmful content after it has been produced by the model; and inference-time protection [55, 20], which intercepts the model's reasoning process by analyzing activation patterns, representation spaces during inference to identify and block unsafe queries before response generation. The Ships [57] also analyzes the internal mechanisms of LVLMs by quantifying the safety contribution of individual

attention heads. However, it does not propose a defense mechanism and exhibits certain limitations, which are discussed in detail in Appendix A.6

Despite these diverse approaches, existing methods often introduce prohibitive computational overhead during deployment that makes them impractical for real-time applications, and even inference-time approaches with relatively lower resource consumption still fail to effectively defend against novel and sophisticated attack patterns, while frequently disrupting the balance between safety and utility with excessive false positives that significantly degrade legitimate user experiences.

2.2 Multi-Head Attention and Hidden States in LVLMs

Multi-Head Attention (MHA) [38] is one of the key innovations enabling LLMs to process sequence data efficiently. The core strength of this mechanism lies in its ability to employ multiple attention heads simultaneously, with each head independently learning to focus on different parts of the input sequence or capture distinct types of semantic relationships. In the Transformer model architecture, hidden states are the outputs of each layer of the model, representing the layer-by-layer representations constructed as the model processes the input sequence. As the input information passes through each layer, the hidden states are continuously updated and enriched, encoding high-dimensional abstract information about the input tokens and their context [20].

3 Amplifying LVLM Safety through Critical Attention Head

This section details our approach, One-head Defense (Oh Defense), to amplify LVLM safety by leveraging intrinsic model capabilities. We find that sensitivity to harmful content concentrates within specific safety-critical attention heads. The Oh Defense first identifies these heads, and monitors how masking a single critical head alters internal representations, measured by its "deflection angle", enables efficient, training-free detection of unsafe queries. Once a threat is detected, a customized response method is triggered to mitigate potential risks.

3.1 Identifying Safety-Critical Attention Heads

We identify safety-critical attention heads through a systematic analysis of how each attention head influences the processing of safe and unsafe content. By selectively masking individual attention heads and measuring the resulting changes in hidden states, we can pinpoint heads that are particularly sensitive to safety.

Representing Model Behavior via Principal Hidden States. We first aim to summarize the model's behavior over a dataset using a compact representation of its hidden states. Given a dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_{|\mathcal{D}|}\}$, we extract the hidden states $h_j \in \mathbb{R}^{1\times d}$ (d is the dimension of hidden states) corresponding to the last input token in the model's final layer for each sample x_j, which is commonly considered to be the final contextualized representation formed after the model has processed all input information and understood the entire prompt [52]. These hidden states are then concatenated into a matrix:

$$H^{\mathcal{D}} = \text{concat}_{\text{row}}(h_1, h_2, \ldots, h_{|\mathcal{D}|}) \in \mathbb{R}^{|\mathcal{D}|\times d}. \tag{1}$$

To obtain a summary representation, we extract the first r column of $H^{\mathcal{D}}$ denoted as:

$$v^{\mathcal{D}} = H^{\mathcal{D}}[:, :r]. \tag{2}$$

This projection captures the salient direction in the hidden space [44]; and empirically, using $r = 1$ is already able to capture the variation in the data while making the calculation more efficient [57]. Note Appendix A.8 provides a detailed justification for the effectiveness of using $r = 1$ in capturing data variation. Thus, $v^{\mathcal{D}} \in \mathbb{R}^{|\mathcal{D}|\times 1}$ serves as our principal behavior vector for the given dataset $\mathcal{D}$.

Quantifying the Impact of Individual Attention Heads. To assess the functional role of each attention head, we introduce a lightweight masking method. For each attention head (l, i), where l is the layer index and i is the head index, we scale its Query weight matrix by a very small coefficient ϵ (typically 10^{-5}) to suppress the head's contribution:

$$\text{head}_i^{\text{mask}} = \text{softmax}\left(\frac{(Q \cdot \epsilon W_i^Q)K_i^\top}{\sqrt{d_k}}\right) V_i, \tag{3}$$

where Q, K, V, and W represent the Query, Key, Value matrices, and weight matrix in the Transformer architecture. Then, the modified attention output is propagated through the MHA mechanism (Equation (4)).

$$\mathrm{MHA}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_i, \ldots, \mathrm{head}_n)W^O, \tag{4}$$

where W^O denotes the output weight matrix. Next, we recalculate the hidden states matrix $H^{\mathcal{D}}_{(l,i)_{\text{mask}}}$ and its principal vector $v^{\mathcal{D}}_{(l,i)_{\text{mask}}}$. To quantify the effect of masking, we measure the *deflection angle* $\theta^{\mathcal{D}}_{(l,i)_{\text{mask}}}$, deflecting from the original principal vector $v^{\mathcal{D}}_{\text{base}}$ to the masked principal vector $v^{\mathcal{D}}_{(l,i)_{\text{mask}}}$. Equation (5) calculates the deflection angle:

$$\theta^{\mathcal{D}}_{(l,i)_{\text{mask}}} = \frac{180}{\pi} \cdot \arccos\left(\frac{(\mathbf{v}^{\mathcal{D}}_{\text{base}})^\top \cdot \mathbf{v}^{\mathcal{D}}_{(l,i)_{\text{mask}}}}{\|\mathbf{v}^{\mathcal{D}}_{\text{base}}\|_2 \cdot \|\mathbf{v}^{\mathcal{D}}_{(l,i)_{\text{mask}}}\|_2}\right). \tag{5}$$

This angle measures how much the model's final-layer representation is shifted by deactivating the head (l, i).

Identifying Safety-Critical Heads via Deflection Differentials. We calculate the deflection angle $\theta^{\mathcal{D}_{\text{safe}}}_{(l,i)}$ and $\theta^{\mathcal{D}_{\text{unsafe}}}_{(l,i)}$ for each attention head (l, i) on the prepared safe and unsafe datasets. Based on these calculated deflection angle, a head (l, i) is considered safety-critical if it satisfies two criteria:

• **Limited Impact on Safe Inputs:** The impact on safe inputs is relatively small. Its deflection angle on safe datasets is less than or equal to the threshold τ:

$$\theta^{\mathcal{D}_{\text{safe}}}_{(l,i)_{\text{mask}}} \leq \tau. \tag{6}$$

This condition is used to filter out heads that have a generally large impact across all input types (including safe inputs), thereby focusing the selection on identifying heads that contribute specifically to safety properties. τ is set at the 80th percentile of these deflection angle across all heads, meaning we effectively exclude the 20% of attention heads exhibiting the highest impact on safe content.

• **Distinction between safe and unsafe Inputs (Top-k Selection):** From the heads that satisfy the first condition, we then select those that most strongly differentiate between safe and unsafe inputs. We calculate the difference in deflection angle:

$$\Delta\theta_{(l,i)_{\text{mask}}} = \theta^{\mathcal{D}_{\text{unsafe}}}_{(l,i)_{\text{mask}}} - \theta^{\mathcal{D}_{\text{safe}}}_{(l,i)_{\text{mask}}}. \tag{7}$$

An attention head (l, i) is selected for a safety head candidates set S_{safe} if $\Delta\theta_{(l,i)_{\text{mask}}}$ ranks among the top-k (here we set $k = 5$) highest values obtained from the set of heads that passed the first condition.

3.2 Input Detection and Response Method

Once safety heads are identified, the subsequent objective is to leverage them for a practical input detection and response. Inputs are classified by comparing their deflection angle at a chosen critical head against a threshold. Unsafe inputs trigger a guided refusal, initiated by anchored output tokens.

Calculating Deflection Angle for Individual Samples and Determining Classification Threshold. To determine if an input is unsafe, its deflection angle must first be calculated and this process differs slightly from the previous: we directly use its base hidden states h_{base} as its principal vector (r is d) and the dimension is $1 \times d$. After masking the attention head (l, i), we obtain the new hidden states $h_{(l,i)_{\text{mask}}}$, the deflection angle of this sample can be represented as follows:

$$\theta_{(l,i)_{\text{mask}}} = \frac{180}{\pi} \cdot \arccos\left(\frac{\mathbf{h}_{\text{base}} \cdot \mathbf{h}^\top_{(l,i)_{\text{mask}}}}{\|\mathbf{h}_{\text{base}}\|_2 \cdot \|\mathbf{h}_{(l,i)_{\text{mask}}}\|_2}\right). \tag{8}$$

With a method to calculate the deflection angle for any sample, the next step is to establish a clear decision boundary θ^*. Specifically, we randomly select an attention head (l^*, i^*) from S_{safe} and calculate the deflection angle for all samples in both $\mathcal{D}_{\text{safe}}$ and $\mathcal{D}_{\text{unsafe}}$ at this head, resulting in two sets of deflection angle: $\Delta_{\text{safe}} = \{\theta^{\text{safe}_1}_{(l^*,i^*)_{\text{mask}}}, \theta^{\text{safe}_2}_{(l^*,i^*)_{\text{mask}}}, \ldots, \theta^{\text{safe}_{|\mathcal{D}_{\text{safe}}|}}_{(l^*,i^*)_{\text{mask}}}\}$ and $\Delta_{\text{unsafe}} = \{\theta^{\text{unsafe}_1}_{(l^*,i^*)_{\text{mask}}}, \theta^{\text{unsafe}_2}_{(l^*,i^*)_{\text{mask}}}, \ldots, \theta^{\text{unsafe}_{|\mathcal{D}_{\text{unsafe}}|}}_{(l^*,i^*)_{\text{mask}}}\}$. By applying kernel density estimation (KDE) to

these sets of deflection angle, we can obtain probability density functions f_{safe} and f_{unsafe}, which are then used to determine the optimal classification threshold θ^*:

$$\theta^* = \arg\min_{\theta} |f_{\text{safe}}(\theta) - f_{\text{unsafe}}(\theta)|. \tag{9}$$

Input Detection and Guided Response Strategy. With the threshold θ^* determined, for new input sample, we calculate its deflection angle at the selected head (l^*, i^*). If this angle exceeds θ^*, the input is classified as unsafe; otherwise, it is considered safe. Upon detecting unsafe input, the traditional method is to directly output a refusal, which often appears rigid [42]. An alternative method is to add a safety prompt to guide the model toward generating safe answers [52, 55]. However, carefully crafted unsafe prompts can sometimes bypass these safety measures. To address these limitations, we propose an improved safety response strategy that not only refuses unsafe requests but also guides the model to generate a reasoned refusal. When unsafe input is detected, we add a safety prompt to the model, requesting it to provide a refusal response and explain the reason. Crucially, we fix the initial tokens of the model's output to a specific phrase like "I cannot". The method steers the model's autoregressive output by initiating its response with a key phrase from our safety prompt, which enforces immediate alignment and sets a robust, safety-aligned trajectory. This significantly hinders jailbreak attempts aiming to override the safety prompt and elicit harmful content (safety prompt we designed and the performance of anchoring tokens are detailed in Appendix A.9).

4 Experimental Evaluation

4.1 Configurations

Datasets. In the experiments, we use VLSafe [11], JailbreakV-28K [31] as the unsafe datasets, which are widely used in existing works [55, 54, 20]. Specifically, VLSafe consists of 1110 unsafe image-text pairs, where the harmful content is explicitly present within the text without any form of disguise. JailbreakV-28K, one of the most comprehensive and widely utilized benchmark datasets for multimodal harmful content currently available, encompasses $28,000$ jailbreak attack instances across 16 attack scenarios, organized into three main categories: LLM Transfer, which involves migrating jailbreaking techniques (originally developed for LLMs) to LVLMs by integrating them with different images; FigStep, which embeds unsafe information directly into images through typography; and Query-Relevant (QR), derived from MM-SafetyBench [28], which utilizes three methods for image generation: Stable Diffusion (SD), Typography and SD + Typography. Meanwhile, we randomly selected 500 samples each from LLaVA-Instruct-80K [25] and ShareGPT4V [7], LLaVA-Instruct-80K comprises 80k instruction samples generated by GPT-4 from COCO images, while ShareGPT4V contains over 100k diverse real-world user dialogues with GPT-4V. Additionally, we selected 50 samples each from VLSafe and LLaVA-Instruct-80K to serve as our $\mathcal{D}^{\text{unsafe}}$ and $\mathcal{D}^{\text{safe}}$ in our method.

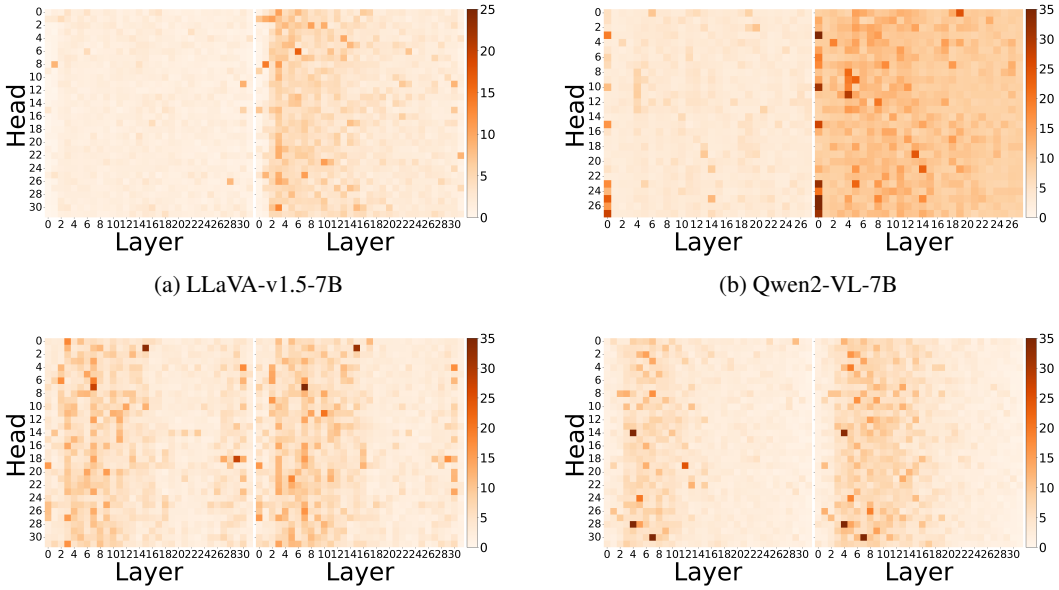
Base LVLM models. In our work, we primarily analyze four well-known LVLMs built upon different base LLM architectures, including LLaVA-v1.5-7B [25], Qwen2-VL-7B [39], Aya-Vision-8B [12], and Phi-3.5-Vision [1], which are also widely used in existing works [54, 20, 55].

Evaluation Model and Metric. For response safety evaluation, we employ LLaMAGuard [18]. LLaMAGuard is a safety-tuned LLM that, like models such as WildGuard [16] and ShieldGemma [50], provides a robust solution by jointly evaluating the prompt and response for harmful content. We choose LLaMAGuard specifically due to its demonstrated stronger performance for safety detection compared to models like GPT-4o [30]. Given a dataset $\mathcal{D}$, we use a commonly used metric, Attack Success Rate (ASR) [31, 48] (Equation (10)).

$$\text{ASR}(M, \mathcal{D}, E) = \frac{1}{|\mathcal{D}|} \sum_{(p,i)\in\mathcal{D}} E(p, M(p,i)) == \text{True}, \tag{10}$$

where M is the LVLM being evaluated, (p, i) is a text/image pair in dataset $\mathcal{D}$, and E is the safety evaluation model. Detection Rate (DR) denotes the proportion of inputs detected as unsafe by the Oh Defense, providing a straightforward measure of our defense's effectiveness.

Baseline. We compare the Oh Defense against four competitive and state-of-the-art approaches: The SAHs [55] and the HiddenDetect [20] for inference-time defense (directly comparable to our method), alongside the widely-used BlueSuffix [54] for input purification and ECSO [15] for post-processing as representative baselines in their respective categories.



(a) LLaVA-v1.5-7B (b) Qwen2-VL-7B

(c) AyaVision-8B (d) Phi-3.5-Vision

Figure 2: Heatmaps showing deflection angle for attention heads across different LVLMs. Each heatmap displays the deflection angle for each attention head when processing $\mathcal{D}^{\text{safe}}$ (left) versus $\mathcal{D}^{\text{unsafe}}$ (right). The language model of Qwen2-VL-7B consists of 28 layers, each with 28 attention heads, while the language models of the others consist of 32 layers, each with 32 attention heads.

4.2 Main Results

We now present the main experimental results evaluating the Oh Defense. These results detail the behavioral differences of attention heads in response to safe versus unsafe inputs, the discriminative capability of a single identified safety head for input classification, and a comparative performance analysis of the Oh Defense against existing defense strategies.

Visualizing Deflection Angle Differences in Attention Heads. For these models, we use the approach in Section 3.1 to identify the safety heads. We first calculate the deflection angle for every attention head. Figure 2 displays the deflection angle for each attention head across different LVLMs, comparing safe versus unsafe inputs. A more intense color signifies a larger deflection angle and it is evident from right-hand heatmaps in each pair that when processing unsafe inputs, a noticeably greater number of attention heads exhibit these larger deflection angle.

Single Head's Discriminative Power for Input Safety. Using our method from Section 3.1, we could identify a set of safety heads candidate S_{safe} for each model, e.g., $\{(8, 2), (10, 23), (5, 5), (3, 10), (5, 28)\}$ for LLaVA-v1.5-7B. Then following the method in Section 3.2, we randomly sample one attention head from S_{safe} as the safety head to detect if the input is unsafe. As shown in Table 1, we observe remarkably high DR across all unsafe datasets, with the Oh Defense achieving near-perfect performance, most over 99% on the VLSafe, QR, FigStep, and LLM Transfer datasets. Crucially, the Oh Defense maintains very low false positive rates on safe datasets, as evidenced by DR of only $2 \sim 9.62\%$ on Instruct-80k and $0 \sim 1.40\%$ on ShareGPT4V. This distinct performance on both safe and unsafe datasets is consistently observed across various LVLM architectures, regardless of their underlying base language models, thereby validating the generalization ability of the Oh Defense. This can be attributed to the fact that the safety head we selected demonstrates high sensitivity to unsafe inputs while containing minimal redundant information, which is the key to its outstanding effectiveness. Further, we present the detection performance of additional five models in Appendix A.7, along with the selection of relevant parameters for all models.

Enhanced Defense Performance Compared to Existing Methods. After detecting unsafe input, we implement defense mechanism that integrates a safety prompt with fixing the initial tokens of the output. As shown in Table 2, the Oh Defense demonstrates superior performance compared to several

Table 1: DR (%) for different LVLMs on various unsafe (VLSafe, QR, Figstep, and LLM Transfer) and safe (Instruct-80k and ShareGPT4V) datasets using a randomly selected safety head in S_{safe}. Note: in practice, we can select the best head using a small validation set such as $\mathcal{D}^{\text{safe}}$ and $\mathcal{D}^{\text{unsafe}}$.

Model	Basic LLM	Head	VLSafe	QR	Figstep	LLM Transfer	Instruct-80k	ShareGPT4V
LLaVA-v1.5-7B	Vicuna-v1.5-7B	(8, 2)	99.91	98.47	100	99.92	2	0
Qwen2-VL-7B	Qwen2-7B	(15, 22)	99.34	94.45	100	88.73	9.62	0
Aya-Vision-8B	Command R7B	(16, 17)	99.34	83.82	98.95	99.67	5	1.40
Phi-3.5-Vision	Phi-3 Mini	(16, 18)	99.91	99.35	93.95	99.92	1.80	0

Table 2: A comparative analysis of ASR (%) for different defense strategies applied to four LVLM architectures against diverse unsafe datasets. We mark the best result in boldface.

Model	Defense	VLSafe	QR	Figstep	LLM Transfer	Average
LLaVA-v1.5-7B	No Defense	80.27	23.86	41.85	66.90	53.22
	ECSO	4.41	6.83	12.40	61.60	21.31
	BlueSuffix	22.07	9.63	41.25	20.09	23.26
	SAHs	0.36	1.40	41.85	41.05	21.17
	HiddenDetect	0.27	1.50	11.70	0.69	3.54
	Oh Defense	**0.19**	**0**	**0**	**0.24**	**0.11**
Qwen2-VL-7B	No Defense	20.36	22.55	32.45	14.79	22.54
	ECSO	4.05	12.53	17.35	21.96	13.97
	BlueSuffix	6.58	11.30	30.25	3.25	12.85
	SAHs	**0.18**	**0.01**	**0**	10.63	2.71
	HiddenDetect	8.74	12.72	0.75	4.57	6.70
	Oh Defense	1.89	1.10	**0**	**3.22**	**1.55**
Aya-Vision-8B	No Defense	9.10	16.67	30.90	59.37	29.01
	ECSO	4.59	15.25	25.30	62.19	26.83
	BlueSuffix	4.50	8.21	30.45	10.97	13.53
	SAHs	8.46	7.11	30.90	37.72	21.05
	HiddenDetect	1.36	**0.13**	14.40	**0.14**	4.01
	Oh Defense	**1.23**	5.11	**0.01**	0.50	**1.71**
Phi-3.5-Vision	No Defense	1.44	0.40	25.95	9.37	9.29
	ECSO	0.85	0.22	12.95	11.86	6.47
	BlueSuffix	3.45	0.28	18.80	2.58	6.28
	SAHs	**0.18**	0.10	4.55	9.30	3.53
	HiddenDetect	1.08	0.35	2.20	1.62	1.31
	Oh Defense	0.28	**0.01**	**1.65**	**0.82**	**0.69**

existing defense strategies across various LVLMs and unsafe datasets, achieving the lowest ASR ($< 2\%$) in the majority of test scenarios. Notably, the Oh Defense completely neutralizes attacks from Figstep, a particularly evasive method that bypasses most prior defenses, where the Oh Defense consistently drives ASR to near-zero levels. These findings suggest that our strategy effectively enhances the robustness of LVLMs against diverse attacks, substantially mitigating the risk of unsafe response generation. The Oh Defense shows similarly excellent performance in defending against adversarial attacks, which is further elaborated in Appendix A.4. For a more detailed comparison between similar methods such as the SAHs and the Ships, see Appendix A.6.

4.3 Further Analysis

After validating what the Oh Defense achieves, we will subsequently delve deeper into why and how it operates, offering a more interpretable and comprehensive perspective on its capabilities.

One Head is Enough. We have found that a single safety head is already effective in accomplishing defensive tasks, which raises the question: could increasing the number of safety heads lead to enhanced defensive performance? Table 3 presents our detection results across various safety head configurations on different models. Interestingly, increasing the quantity of safety heads did not enhance the model's overall defensive capabilities; in fact, it sometimes degrades performance (e.g., on QR and Instruct-80k). This can be attributed to the introduction of inconsistent or less discriminative attention patterns from additional heads, which may blur the decision boundary and reduce overall robustness. In contrast, a single well-identified safety head provides a more coherent and focused understanding of unsafe features, yielding better generalization across diverse scenarios.

These findings suggest that one safety head is sufficient, and even preferable, for achieving effective and interpretable defense.

Table 3: Comparison of DR (%) using different numbers of safety heads on multiple datasets and various models.

Model	Head Number	θ^*	VLSafe	QR	Figstep	LLM Transfer	Instruct-80k	ShareGPT4V
LLaVA-v1.5-7B	Top-1	2.16	99.91	98.47	100	99.92	2	0
	Top-3	4.54	99.36	97.17	100	99.90	1.20	0
	Top-5	6.21	98.46	99.33	100	99.95	2.60	0
Qwen2-VL-7B	Top-1	1.22	99.34	94.45	100	88.73	9.62	0
	Top-3	3.02	89.81	94.17	100	88.20	4.40	0.60
	Top-5	3.84	86.12	70.10	100	75.67	5.40	0.40
Aya-Vision-8B	Top-1	2.30	99.34	83.82	98.95	99.67	5	1.40
	Top-3	3.41	96.48	92.33	100	98.95	10.40	1.60
	Top-5	4.44	94.68	99.67	100	100	13	4.20
Phi-3.5-Vision	Top-1	3.50	99.91	99.35	93.95	99.92	1.80	0
	Top-3	6.65	99.72	98.17	36.50	99.20	0.60	0
	Top-5	7.23	99.81	99	55	99.65	1	0

Safety Mechanisms in LVLMs with Similar Base LLMs. Figure 2 depicts that the distribution of safety attention heads differs across LVLMs built upon different base language models, for instance, Qwen2-VL-7B shows safety-critical heads concentrated more in later layers, while AyaVision-8B displays them more prominently in early and middle layers. To investigate this phenomenon more deeply, we analyze three well-known models, all based on the LLaMA family [47]: LLaVA-v1.5-7B, ShareGPT4V-7B, and Magma-8B. Specifically, both LLaVA and ShareGPT4V are built upon Vicuna-v1.5-7B (fine-tuned from LLaMA-2-7B), while Magma-8B adopts LLaMA-3-8B as its base. As shown in Figure 3, although ShareGPT4V-7B exhibits generally larger deflection angle than LLaVA-v1.5-7B on $\mathcal{D}^{\text{unsafe}}$, due to differences in fine-tuning, the two models display remarkably consistent safety-related trends across layers. For example, in both cases, deflection angle increase sharply from layer 0 to 3, peak around layer 3, and gradually decline thereafter. Magma-8B, despite being based on a similar LLM (LLaMA-3), follows a similar trajectory. This suggests that while the exact positions and sensitivities of safety heads may differ across models, the overall pattern of safety-relevant attention dynamics remains structurally consistent across base LLMs.

Query vs. Value Weight Matrix Masking Effects. In our method, we analyze attention heads by masking them through scaling their Query weight matrices. While scaling either Query or Key matrices similarly affects attention scores ($QK^\top$), consequently altering the distribution of attention weights. Scaling the Value weight, however, only reduces the contribution of weighted values after attention computation without disrupting the underlying attention pattern. When scaling the Query matrix with an extremely small coefficient, the Query-Key dot products become significantly smaller, causing the softmax function to produce a more uniform attention weight distribution. This uniformity effectively suppresses the distinctive patterns learned by the attention head, more thoroughly deactivating its function compared to Value scaling, which merely attenuates output magnitude. As observed in Figure 4, scaling the Query weight matrix produces larger average deflection angle than scaling the Value matrix, providing clearer signals for identifying attention heads critical to safety performance. The most significant differences appearing in the earliest layers, which suggests that the initial layers perform the crucial task of extracting low-level features and establishing preliminary connections. The model likely begins identifying safety-related foundational elements, such as potentially unsafe vocabulary or trigger phrases, at these shallow layers. Complementary results from scaling the Value matrix are included in Appendix A.10.

Choice of Hidden Layer for Monitoring Safety Signals. According to our method in Section 3.1, after masking attention heads, we choose to observe changes in the hidden states of the final layer, though this is not the only possible selection. Figure 5 illustrates the distribution of deflection angle between $\mathcal{D}^{\text{safe}}$ and $\mathcal{D}^{\text{unsafe}}$ across different hidden layers after masking three different safety heads in the LLaVA-v1.5-7B. The visualization shows that in the later layers of the model (approximately after layer 20), the model becomes capable of effectively distinguishing between safe and harmful content. This reflects that the safety functionality of the model has largely completed its identification and processing of unsafe information in these deeper layers. At this stage, the information aggregated and

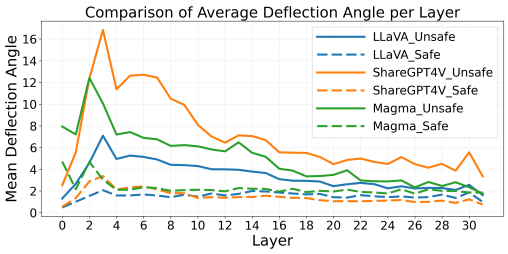


Figure 3: Comparison of average deflection angle across different layers for three LVLMs (LLaVA-v1.5-7B, ShareGPT4V-7B, and Magma-8B) when processing $\mathcal{D}^{\text{safe}}$ and $\mathcal{D}^{\text{unsafe}}$.

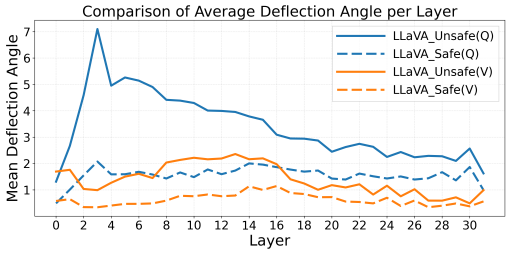
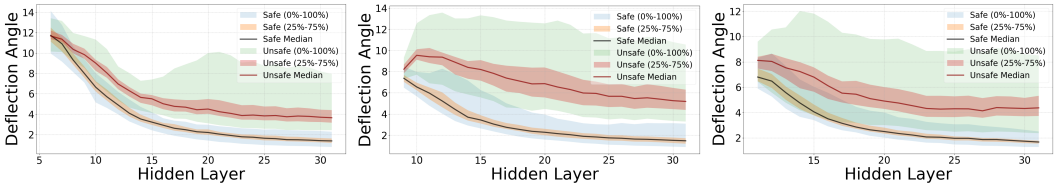


Figure 4: Comparison of average deflection angle when masking different weight matrices (Query vs. Value) across model layers in LLaVA-v1.5-7B.



(a) (5, 5) (b) (8, 2) (c) (10, 23)

Figure 5: Distribution of deflection angle between $\mathcal{D}^{\text{safe}}$ and $\mathcal{D}^{\text{unsafe}}$ across different hidden layers after masking three different safety attention heads (5, 5), (8, 2), and (10, 23) in LLaVA-v1.5-7B.

processed by the deeper layers is sufficient to enable the model to make final discriminations and distinctions between different types of content. Further model results are available in Appendix A.11. A more comprehensive investigation into factors affecting this experiment is presented in Appendix A.8, including analyses of time efficiency, hyperparameter selection, metric ablations, etc.

5 Conclusion

In this work, we address the critical issue of safety alignment degradation in LVLMs. We introduce a novel method, One-head Defense (Oh Defense), that leverages the intrinsic multi-head attention mechanisms within LVLMs to enhance safety. By identifying safety-critical attention heads respond distinctly to unsafe inputs, we show a single critical head can effectively maintain model security without requiring retraining or external modules. Meanwhile, once an unsafe input is detected, we activate a prompt-level defense strategy that appends a safety prefix and anchors the first few output tokens, thereby steering the model away from unsafe generations. This work not only provides a practical defense method but also offers valuable insights into the inherent safety structures within LVLMs, paving the way for future research into lightweight and interpretable safety solutions.

Limitations. While the Oh Defense demonstrates strong performance and efficiency, we acknowledge certain limitations. Our findings suggest that the identified critical head is model-dependent, necessitating a tailored identification process for each distinct LVLM to ensure optimal performance. Moreover, potential deployment risks, such as misuse for overconfident content filtering or fully automated moderation without human oversight, remain unexplored. Although the Oh Defense achieves low false positive rates on benign datasets ($< 5\%$ on most benchmarks, as shown in Table 1), meaning most non-malicious tasks proceed normally, this safety-first design inherently involves a trade-off between utility and risk mitigation. In edge cases, prioritizing safety may lead to conservative behavior that impacts response quality.

Acknowledgments

Junhao Xia, Haotian Zhu, and Shuchao Pang are supported by the National Natural Science Foundation of China (No. 62206128) and the National Key R&D Program of China (No. 2023YFB2703900). Yongbin Zhou is supported by the National Natural Science Foundation of China (No. U2336205).

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, 2023.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Vivien Cabannes, Charles Arnal, Wassim Bouaziz, Xingyu Alice Yang, Francois Charton, and Julia Kempe. Iteration head: A mechanistic study of chain-of-thought. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[6] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramèr, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[7] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer, 2024.

[8] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, Li Yuan, Yu Qiao, Dahua Lin, Feng Zhao, and Jiaqi Wang. ShareGPT4video: Improving video understanding and generation with better captions. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[9] Xingwu Chen, Lei Zhao, and Difan Zou. How transformers utilize multi-head attention in in-context learning? a case study on sparse linear regression. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[10] Yangyi Chen, Hongcheng Gao, Ganqu Cui, Fanchao Qi, Longtao Huang, Zhiyuan Liu, and Maosong Sun. Why should adversarial perturbations be imperceptible? rethink the research paradigm in adversarial nlp. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11222–11237, 2022.

[11] Yangyi Chen, Karan Sikka, Michael Cogswell, Heng Ji, and Ajay Divakaran. Dress: Instructing large vision-language models to align and interact with humans via natural language feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14239–14250, 2024.

[12] John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, et al. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*, 2024.

[13] Yi Ding, Bolian Li, and Ruqi Zhang. Eta: Evaluating then aligning safety of vision language models at inference time. In *The Thirteenth International Conference on Learning Representations*, 2025.

[14] Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. Figstep: Jailbreaking large vision-language models via typographic visual prompts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 23951–23959, 2025.

[15] Yunhao Gou, Kai Chen, Zhili Liu, Lanqing Hong, Hang Xu, Zhenguo Li, Dit-Yan Yeung, James T Kwok, and Yu Zhang. Eyes closed, safety on: Protecting multimodal llms via image-to-text transformation. In *European Conference on Computer Vision*, pages 388–404. Springer, 2024.

[16] Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[17] Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic jailbreak of open-source LLMs via exploiting generation. In *The Twelfth International Conference on Learning Representations*, 2024.

[18] Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

[19] Jiaming Ji, Xinyu Chen, Rui Pan, Han Zhu, Conghui Zhang, Jiahao Li, Donghai Hong, Boyuan Chen, Jiayi Zhou, Kaile Wang, et al. Safe rlhf-v: Safe reinforcement learning from human feedback in multimodal large language models. *arXiv preprint arXiv:2503.17682*, 2025.

[20] Yilei Jiang, Xinyan Gao, Tianshuo Peng, Yingshui Tan, Xiaoyong Zhu, Bo Zheng, and Xiangyu Yue. HiddenDetect: Detecting jailbreak attacks against multimodal large language models via monitoring hidden states. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14880–14893, Vienna, Austria, July 2025. Association for Computational Linguistics.

[21] Seongyun Lee, Geewook Kim, Jiyeon Kim, Hyunji Lee, Hoyeon Chang, Sue Hyun Park, and Minjoon Seo. How does vision-language adaptation impact the safety of vision language models? In *The Thirteenth International Conference on Learning Representations*, 2025.

[22] Li Li, Jiawei Peng, Huiyi Chen, Chongyang Gao, and Xu Yang. How to configure good in-context sequence for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26710–26720, 2024.

[23] Yifan Li, Hangyu Guo, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. Images are achilles' heel of alignment: Exploiting visual vulnerabilities for jailbreaking multimodal large language models. In *European Conference on Computer Vision*, pages 174–189. Springer, 2024.

[24] Zhiyuan Li, Dongnan Liu, Chaoyi Zhang, Heng Wang, Tengfei Xue, and Weidong Cai. Enhancing advanced visual reasoning ability of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1915–1929. Association for Computational Linguistics, November 2024.

[25] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[26] Qin Liu, Chao Shang, Ling Liu, Nikolaos Pappas, Jie Ma, Neha Anna John, Srikanth Doss, Lluis Marquez, Miguel Ballesteros, and Yassine Benajiba. Unraveling and mitigating safety alignment degradation of vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3631–3643, Vienna, Austria, July 2025. Association for Computational Linguistics.

[27] Quan Liu, Zhenhong Zhou, Longzhu He, Yi Liu, Wei Zhang, and Sen Su. Alignment-enhanced decoding: Defending jailbreaks via token-level adaptive refining of probability distributions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2802–2816, 2024.

[28] Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *European Conference on Computer Vision*, pages 386–403. Springer, 2024.

[29] Zhendong Liu, Yuanbi Nie, Yingshui Tan, Xiangyu Yue, Qiushi Cui, Chongjun Wang, Xiaoyong Zhu, and Bo Zheng. Safety alignment for vision language models. *arXiv preprint arXiv:2405.13581*, 2024.

[30] AI @ Meta Llama Team. The llama 3 herd of models, 2024.

[31] Weidi Luo, Siyuan Ma, Xiaogeng Liu, Xiaoyu Guo, and Chaowei Xiao. Jailbreakv: A benchmark for assessing the robustness of multimodal large language models against jailbreak attacks. In *First Conference on Language Modeling*, 2024.

[32] Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[33] 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. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16012–16027. Association for Computational Linguistics, November 2024.

[34] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI conference on artificial intelligence*, pages 21527–21536, 2024.

[35] Jie Ren, Qipeng Guo, Hang Yan, Dongrui Liu, Quanshi Zhang, Xipeng Qiu, and Dahua Lin. Identifying semantic induction heads to understand in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6916–6932. Association for Computational Linguistics, August 2024.

[36] Mustafa Shukor and Matthieu Cord. Implicit multimodal alignment: On the generalization of frozen llms to multimodal inputs. *Advances in Neural Information Processing Systems*, 37:130848–130886, 2024.

[37] Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, et al. A strongreject for empty jailbreaks. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[39] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[40] Pengyu Wang, Dong Zhang, Linyang Li, Chenkun Tan, Xinghao Wang, Mozhi Zhang, Ke Ren, Botian Jiang, and Xipeng Qiu. InferAligner: Inference-time alignment for harmlessness through cross-model guidance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10460–10479. Association for Computational Linguistics, November 2024.

[41] Yanbo Wang, Jiyang Guan, Jian Liang, and Ran He. Do we really need curated malicious data for safety alignment in multi-modal large language models? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19879–19889, 2025.

[42] Yihan Wang, Zhouxing Shi, Andrew Bai, and Cho-Jui Hsieh. Defending LLMs against jailbreaking attacks via backtranslation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16031–16046. Association for Computational Linguistics, August 2024.

[43] Yu Wang, Xiaogeng Liu, Yu Li, Muhao Chen, and Chaowei Xiao. Adashield: Safeguarding multimodal large language models from structure-based attack via adaptive shield prompting. In *European Conference on Computer Vision*, pages 77–94. Springer, 2024.

[44] Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *Proceedings of the 41st International Conference on Machine Learning*, pages 52588–52610, 2024.

[45] Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, Ruoxi Jia, Bo Li, Kai Li, Danqi Chen, Peter Henderson, and Prateek Mittal. SORRY-bench: Systematically evaluating large language model safety refusal. In *The Thirteenth International Conference on Learning Representations*, 2025.

[46] Yue Xu, Xiuyuan Qi, Zhan Qin, and Wenjie Wang. Defending jailbreak attack in vlms via cross-modality information detector. *arXiv e-prints*, pages arXiv–2407, 2024.

[47] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal ai agents. *arXiv preprint arXiv:2502.13130*, 2025.

[48] Mang Ye, Xuankun Rong, Wenke Huang, Bo Du, Nenghai Yu, and Dacheng Tao. A survey of safety on large vision-language models: Attacks, defenses and evaluations. *arXiv preprint arXiv:2502.14881*, 2025.

[49] Zonghao Ying, Aishan Liu, Tianyuan Zhang, Zhengmin Yu, Siyuan Liang, Xianglong Liu, and Dacheng Tao. Jailbreak vision language models via bi-modal adversarial prompt. *IEEE Transactions on Information Forensics and Security*, 2025.

[50] Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, et al. Shieldgemma: Generative ai content moderation based on gemma. *arXiv preprint arXiv:2407.21772*, 2024.

[51] Xiaoyu Zhang, Cen Zhang, Tianlin Li, Yihao Huang, Xiaojun Jia, Xiaofei Xie, Yang Liu, and Chao Shen. A mutation-based method for multi-modal jailbreaking attack detection. *CoRR*, 2023.

[52] Qinyu Zhao, Ming Xu, Kartik Gupta, Akshay Asthana, Liang Zheng, and Stephen Gould. The first to know: How token distributions reveal hidden knowledge in large vision-language models? In *European Conference on Computer Vision*, pages 127–142, 2024.

[53] Wei Zhao, Zhe Li, Yige Li, Ye Zhang, and Jun Sun. Defending large language models against jailbreak attacks via layer-specific editing. In *EMNLP 2024-2024 Conference on Empirical Methods in Natural Language Processing, Findings of EMNLP 2024*, pages 5094–5109. Association for Computational Linguistics (ACL), 2024.

[54] Yunhan Zhao, Xiang Zheng, Lin Luo, Yige Li, Xingjun Ma, and Yu-Gang Jiang. Bluesuffix: Reinforced blue teaming for vision-language models against jailbreak attacks. In *The Thirteenth International Conference on Learning Representations*, 2025.

[55] Ziwei Zheng, Junyao Zhao, Le Yang, Lijun He, and Fan Li. Spot risks before speaking! unraveling safety attention heads in large vision-language models. *arXiv preprint arXiv:2501.02029*, 2025.

[56] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

[57] Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, Kun Wang, Yang Liu, Junfeng Fang, and Yongbin Li. On the role of attention heads in large language model safety. In *The Thirteenth International Conference on Learning Representations*, 2025.

[58] Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy Hospedales. Safety fine-tuning at (almost) no cost: a baseline for vision large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 62867–62891, 2024.

A Appendix

A.1 Safety Degradation of LVLMs

Table 4: ASR (%) comparison across different safety benchmarks for Vicuna vs. LLaVA models with various image types (Blank, Noise, and SD)

	Maliciousinstruct [17]	SorryBench [45]	StrongReject [37]	JailbreakBench [6]
Vicuna	59	39.77	28.43	41
LLaVA+Blank	70	46.82	40.26	51
LLaVA+Noise	70	49	39.62	53
LLaVA+SD	71	55	48.56	57

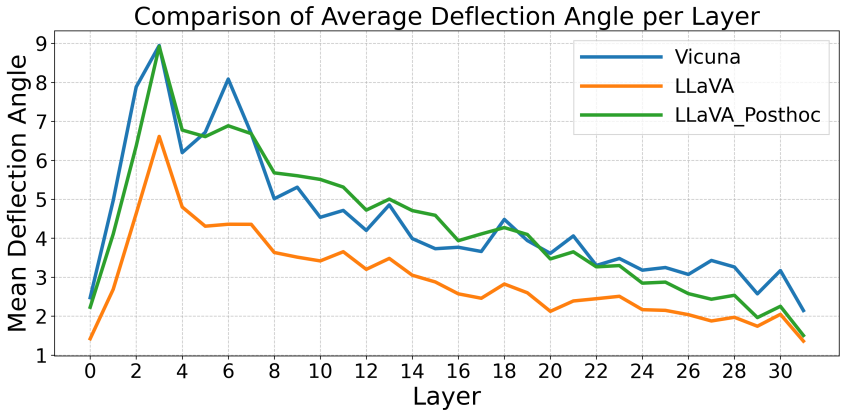


Figure 6: Comparison of average deflection angle per layer across three models: Vicuna-v1.5-7B, LLaVA-v1.5-7B, and LLaVA-v1.5-7B-Posthoc (safety-fine-tuned).

We examine the safety vulnerabilities of LVLMs, focusing specifically on LLaVA-v1.5-7B, which is built upon the Vicuna-v1.5-7B language model. Our investigation employs four text-based unsafe datasets to evaluate ASR. To construct multimodal input scenarios, we augment these unsafe texts with various image types: black images, noise images, and semantically relevant images generated via Stable Diffusion. These text-image combinations were then fed into LLaVA-v1.5-7B to measure the resulting ASR. Table 4 reveals a concerning trend: compared to the text-only Vicuna-v1.5-7B model, LLaVA-v1.5-7B demonstrate significantly higher ASR when processing inputs containing images, even when these images bear no semantic relation to the text. More strikingly, the inclusion of semantically relevant images leads to an even greater increase in ASR.

To better understand the mechanisms underlying this safety degradation, we analyze the attention behavior across three models: LLaVA-v1.5-7B, its base language model Vicuna-v1.5-7B, and a safety-fine-tuned variant, LLaVA-v1.5-7B-Posthoc [58]. Using the MaliciousInstruct [17] dataset, we compute the average deflection angle of attention heads at each layer. Figure 6 shows that across all layers, the standard (non-safety-tuned) LLaVA consistently exhibits smaller average deflection angle compared to both Vicuna and the safety-fine-tuned model. This indicates that in multimodal settings, the attention distribution of non-safety-tuned LVLMs is subject to certain constraints, which may contribute to their observed safety vulnerabilities. Nonetheless, despite these altered attention patterns, our analysis confirms that the attention heads still retain substantial informational content, offering potential utility for future safety analysis or detection tasks.

A.2 Performance Across Harmful Content Categories

While the Oh Defense is designed for detecting safe/unsafe content in LVLMs and cannot yet identify different types of harmful content, it can provide graded risk scores based on the magnitude of

deflection angle. Moreover, the Oh Defense demonstrates good performance against harmful inputs of any category. For instance, the JailbreakV-28K dataset used in this paper contains 16 categories. To explore category-specific defenses, we present the defensive performance of Qwen2-VL-7B and LLaVA-v1.5-7B across different categories on this dataset as follows:

Table 5: Performance (ASR %) of LLaVA-v1.5-7B and Qwen2-VL-7B on the JailbreakV-28K dataset across 16 harmful content categories.

Model	AA	B	CAC	EH	F	GD	HS	HC	IA	M	PH	PS	PV	TUA	UB	V
LLaVA	0.08	0.13	0.00	0.20	0.21	0.34	0.14	0.08	0.18	0.13	0.15	0.00	0.00	0.00	0.36	0.42
Qwen2-VL	2.31	4.19	0.19	2.23	1.78	2.15	3.61	2.67	1.88	1.96	3.22	4.62	2.01	4.73	3.69	1.33

As can be seen, the Oh defense demonstrates robust and consistent protective effectiveness across all categories of harmful datasets. Notably, each category maintains an ASR below 5%. Furthermore, the performance remains remarkably stable across different threat categories, with minimal variation in ASR between categories.

A.3 Oh Defense on Text-Only LLMs

We further supplemented our evaluation with the detection performance of the Oh Defense on two text-only LLMs: Llama2-7B and Phi-3.5-Mini, which are widely used and are the backbone LLMs of the LVLMs used in our paper. We selected three harmful datasets, MaliciousInstruct [17], JailbreakBench [6], and AdvBench [10], along with IFEval [56] as the benign dataset since they are commonly adopted benchmarks for evaluating both the safety and generative capabilities of LLMs. The results are presented as follows:

Table 6: Detection performance (DR %) of Oh Defense on LLaMA2-7B and Phi-3.5-Mini across three harmful datasets and one benign dataset.

Model	Head	Maliciousinstruct	JailbreakBench	AdvBench	IFEval
Llama2-7B	(31, 17)	98	93	95.96	6.70
Phi-3.5-Mini	(16, 18)	98	94	97.97	4.81

As can be observed, the Oh Defense also achieves solid performance on text-only LLMs.

A.4 Oh Defense Against Adversarial Attacks

Table 7: Performance against adversarial attacks on the LLaVA-v1.5-7B model using JailbreakBench [6] under different perturbation constraints ϵ.

	VisualAdv				BAP			
ϵ	16/255	32/255	64/255	unconstrained	16/255	32/255	64/255	unconstrained
ASR (No Defense)	50	60	70	66	44	51	82	60
DR	100	100	100	100	100	100	100	100
ASR	0	0	0	0	0	0	0	0

We further compare the Oh Defense against two adversarial attacks: VisualAdv [34], which targets LVLMs by optimizing adversarial images, and BAP [49], an extension of VisualAdv that additionally performs coordinated text optimizations. We evaluate the Oh Defense under these attacks on LLaVA-v1.5-7B using the JailbreakBench [6] benchmark. As shown in Table 7, the Oh Defense effectively detects both standard adversarial attacks [34] and dual-modal attacks [49], achieving an ASR of 0 across all evaluated scenarios.

Table 8: Average deflection angle comparison across different attack types (VLSafe, QR, FigStep, LLM Transfer, VisualAdv, BAP).

Attack	VLSafe	QR	FigStep	LLM Transfer	VisualAdv	BAP
Average Deflection Angle	4.99	6.91	3.25	5.45	9.82	9.15

To evaluate the Oh Defense, we compute the average deflection angle induced by LLaVA-v1.5-7B when processing inputs from four unsafe datasets and two adversarial attack sets. As shown in Table 8, adversarial attack samples yield significantly higher average deflection angle compared to those from the other unsafe datasets. This suggests that when security-relevant attention heads are masked, adversarial attacks can more readily compromise the model, leading to greater perturbations in the hidden states. These results validate the effectiveness of the Oh Defense in detecting and mitigating strong adversarial attacks.

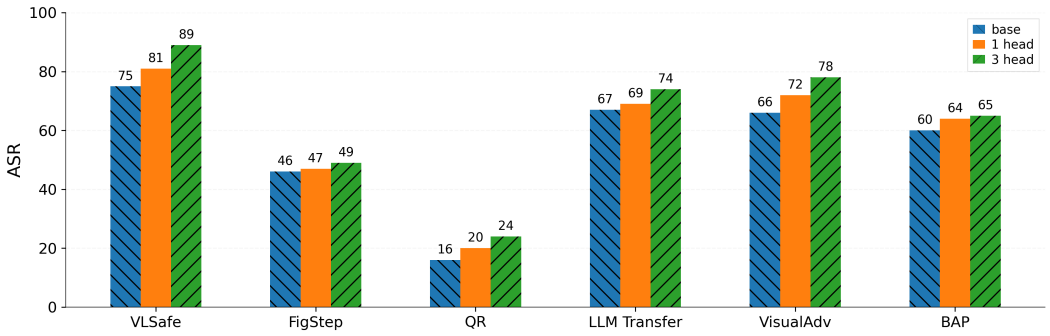


Figure 7: ASR (%) comparison for LLaVA-v1.5-7B across different attack types when masking 0 (base), 1 (8, 2), or 3 ((8, 2),(10, 23),(5, 5)) safety attention heads.

In our earlier experiments, we observed that masking specific attention heads leads to noticeable changes in the model's internal hidden states. To further investigate the impact of these changes, we evaluate the ASR under various attack methods after masking either one or three identified safety attention heads. As shown in Figure 7, masking even a single safety head leads to increased ASR across different attack types, with more pronounced effects when three heads are masked. This suggests that outputs previously deemed safe can become vulnerable to successful attacks once these critical safety heads are disabled. For instance, in the cases of VLSafe and VisualAdv, masking three safety heads increases the ASR by 14% and 12%, respectively. This substantial rise strongly underscores the importance of the identified safety attention heads in preserving model safety and defending against adversarial threats.

A.5 Oh Defense Against Adaptive Attacks

A natural concern regarding our head selection mechanism is whether an adversary could deliberately craft inputs to manipulate or suppress the top-k safety-critical attention heads identified through the Oh Defense, thereby evading detection. To evade the Oh Defense, an attacker would need to satisfy two conditions: (1) precise knowledge of which attention heads are safety-critical for the target model; and (2) the ability to craft inputs that suppress these heads without being flagged by alternative safety heads. These requirements lead to distinct challenges in different threat models:

• **Black-box settings:** In typical API-only deployment scenarios, attackers lack access to internal model parameters or attention head activations. To the best of our knowledge, no existing black-box method can reliably infer the status or identity of safety-critical heads from output observations alone. Thus, the Oh Defense remains highly robust in such practical settings.

• **White-box settings:** While full model access theoretically enables more powerful attacks (e.g., backdoors or direct parameter tampering), we note that even in this extreme case, the Oh Defense retains effectiveness. To evaluate this, we conducted a controlled experiment on a modified LLaVA-

v1.5-7B model in which the original top-5 safety heads were forcibly suppressed (by zeroing their weight matrices). Despite this, the Oh Defense can identify a new set of top-5 safety heads: $\{(3, 19), (14, 27), (13, 7), (4, 21), (5, 24)\}$. Using a randomly selected head from this new set, (13, 7), we evaluated detection rates across multiple benchmarks:

Table 9: Detection performance (DR %) of Oh Defense on LLaVA-v1.5-7B using head (13, 7).

Model	Head	VLSafe	Figstep	QR	LLM Transfer	Instruct-80K	ShareGPT4V
LLaVA-v1.5-7B	(13, 7)	98.91	100	98.50	99.75	3.60	1.60

Although performance on benign instruction-following datasets (e.g., Instruct-80k, ShareGPT4V) slightly decreased, as expected when safety mechanisms are active, the model maintains near-perfect detection on jailbreak and red-teaming benchmarks. This confirms that the Oh Defense can adaptively re-identify safety-critical heads even when the original ones are compromised.

A.6 Comparsion with SAHs and Ships/Sahara

While both the Oh Defense and the SAHs explore the capability of attention heads for safe/unsafe content detection, we observe differences between the Oh Defense and the SAHs in the approach and capabilities.

• **The differences on the approaches:** The SAHs rely on linear probes and trains binary classifiers on attention head activations, requiring supervised learning for detection. In contrast, the Oh Defense adopts a training-free methodology based on principal behavior vectors and deflection angle measurements, avoiding reliance on additional parameters or training data as required by the SAHs.

• **The differences on the detection capabilities:** The Oh Defense demonstrates significant practical advantages. Our approach uses only a single attention head compared to the SAHs' multiple heads (typically 16-32), while achieving superior performance across various attack scenarios. Note that, the Oh defense outperforms the SAHs against challenging attacks like FigStep, see Table 2 for details. The training-free nature also provides better transferability across different models and datasets (though attention heads are selected by VLSafe, the detection capability is demonstrated in Table 1 and Table 2 across various jailbreak datasets), as we don't rely on dataset-specific supervised training that may not generalize well.

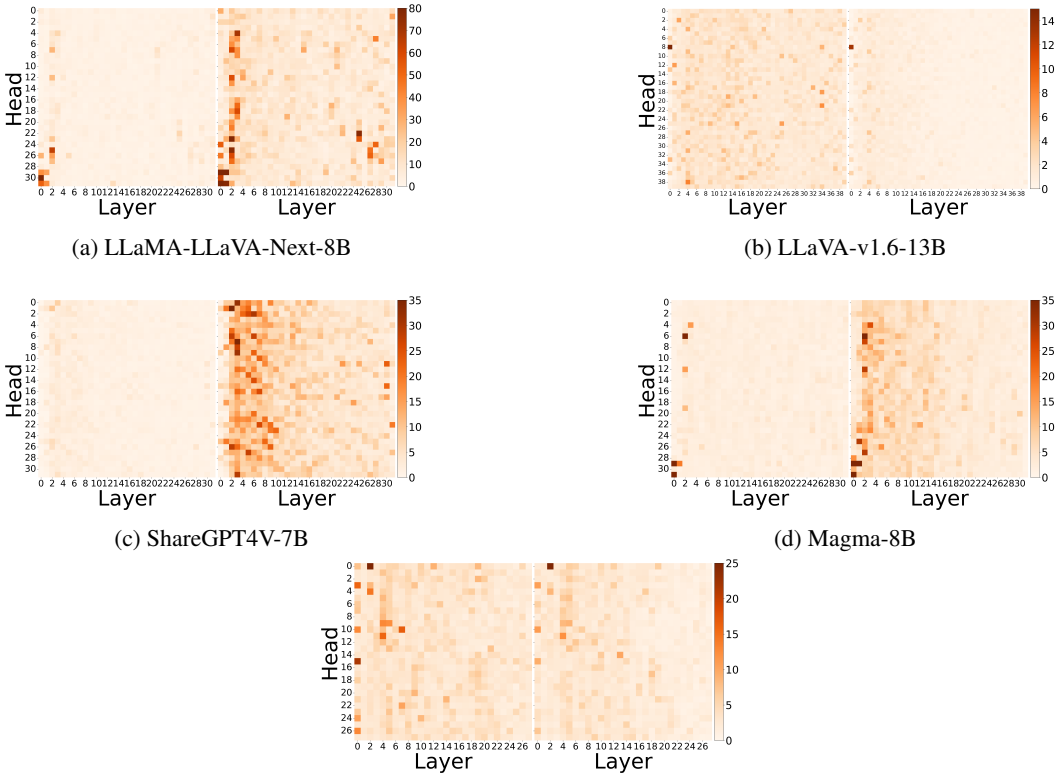
The Ships/Sahara only validates their "safety heads" on harmful datasets. Figure 2 reveals that attention heads showing large deflection angle on unsafe data often exhibit similarly large deflection angle on safe data, particularly in early layers. So these are not true safety heads but rather heads sensitive to general patterns or input structures. We introduce a novel paradigm requiring heads to demonstrate differential behavior between safe and unsafe inputs. Our "Limited Impact on Safe Inputs" (Equation 6) criterion ensures identification of truly safety-critical heads. To demonstrate this critical difference, although the Ships/Sahara does not propose a defense method, we apply the same detection framework to use heads selected by both methods on Qwen2-VL-7B. This ensures we can fairly compare the quality of head selection strategies rather than detection mechanisms. Using KL-based scoring on harmful inputs, the Ships identified the top-performing head as (0, 7), whereas the Oh Defense selected the safety-critical head (15, 22). The results are as follows:

Table 10: Detection performance (DR %) of Ships and Oh Defense on Qwen2-VL-7B.

Method	Head	VLSafe	Figstep	QR	LLM Transfer	Instruct-80K	ShareGPT4V
Ships	(0, 7)	65.50	10	6	19.85	10	0
Oh Defense	(15, 22)	99.34	100	94.45	88.73	9.62	0

The comparison accurately reflects the performance difference between the two head selection approaches under identical detection conditions. It can be observed that when (0, 7) is used as the safety head for detection, the detection rates on FigStep, QR, and LLM Transfer are all below 20%, which indicates that the head selected by the Ships cannot serve as a safety head for detection under parallel settings.

A.7 Performance on Other LVLMs and Detailed Parameter Setting



(a) LLaMA-LLaVA-Next-8B

(b) LLaVA-v1.6-13B

(c) ShareGPT4V-7B

(d) Magma-8B

(e) Qwen2.5-VL-7B

Figure 8: Heatmaps showing deflection angle for attention heads across different LVLMs. Each heatmap displays the deflection angle for each attention head when processing $\mathcal{D}^{\text{safe}}$ (left) versus $\mathcal{D}^{\text{unsafe}}$ (right). The language model of Qwen2.5-VL-7B consists of 28 layers, each with 28 attention heads, LLaVA-v1.6-13B consists of 40 layers, each with 40 attention heads, while the language models of the other models consist of 32 layers, each with 32 attention heads.

Table 11: Performance evaluation metrics for safety detection across different LVLM models, showing each model's base LLM, identified safety head coordinates, classification threshold (θ^*), and detection rate on various datasets.

Model	Basic LLM	Head	θ^*	VLSafe	JailBreakV_28K	Instruct-80k	ShareGPT4V
Llama3-LLaVA-next-8B	Llama-3-8B	(9, 24)	2.41	96.12	89.51	15	0.01
LLaVA-v1.6-13B	Vicuna-13B-v1.5	(4, 16)	3	90.36	98.70	6	14.8
ShareGPT4V-7B	Vicuna-v1.5-7B	(9, 4)	2.74	96.22	99.15	1.60	0.20
Magma-8B	Llama-3-8B	(2, 2)	5.4	97.21	98.81	0.96	4.60
Qwen2.5-VL-7B	Qwen2.5-VL-7B	(13, 14)	1.33	98.56	89.41	6	0.8

In Figure 8, we present heatmaps of attention head deflection angle for various models on both $\mathcal{D}_{\text{safe}}$ and $\mathcal{D}_{\text{unsafe}}$, offering preliminary insights into how attention heads respond differently to distinct input types. Building on this analysis, Table 11 identifies a representative safety attention head for each model, along with its corresponding classification threshold θ^*, and reports the detection performance of the Oh Defense on both safe and unsafe datasets. Together, these results provide strong evidence for the generality and effectiveness of the Oh Defense across multiple LVLMs.

Additionally, Table 12 summarizes the relevant thresholds used across all models discussed in this work. Notably, these thresholds, such as those for identifying safety heads, are not fixed and can be further explored or optimized.

Table 12: Comprehensive threshold parameters for various LVLMs, including τ, identified safety head coordinates, and classification threshold (θ^*) used in the safety detection system.

Model	τ	Head	θ^*
LLaVA-v1.5-7B	1.96	(8, 2)	2.16
Qwen2-VL-7B	4.29	(15, 22)	1.22
Aya-Vision-8B	6.47	(16, 17)	2.3
Phi-3.5-Vision	5.09	(16, 18)	3.5
Llama3-LLaVA-Next-8B	3.62	(9, 24)	2.41
LLaVA-v1.6-13B	2.30	(4, 16)	3
ShareGPT4V-7B	1.95	(9, 4)	2.74
Magma-8B	2.53	(2, 2)	5.4
Qwen2.5-VL-7B	4.31	(13, 14)	1.33

Regarding the critical classification threshold θ^*, which is used to distinguish between safe and unsafe inputs based on specific safety attention heads, we initially estimate its value via KDE. However, because the unsafe content in the VLSafe dataset is relatively explicit and easily detected by the model's internal mechanisms, we typically adjust θ^* downward to enhance robustness against more subtle attacks. Conversely, for models requiring stricter criteria for safety, θ^* can be adjusted upward. This flexibility highlights the tunable nature of θ^* in balancing detection sensitivity and robustness.

A.8 Impact of Different Factors

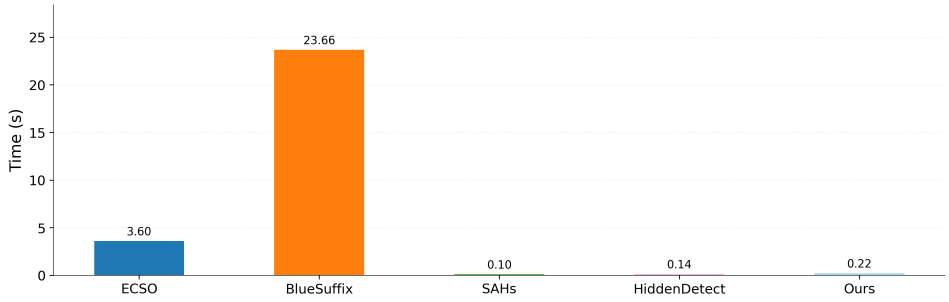


Figure 9: Time Costs in Various Defense Methods.

We compare the time efficiency for processing a single sample across different defense methods applied to the LLaVA-v1.5-7B model. As shown in Figure 9, the input purification method, BlueSuffix, and the post-processing method, ECSO, incur prohibitively high time costs, rendering them infeasible for practical deployment. Compare with the SAHs, although our method requires one additional forward pass, the resulting increase in processing time is well within acceptable limits.

All tasks can be completed on a single NVIDIA RTX 4090 24G GPU and 16vCPU Intel(R) Xeon(R) Gold 6430 on the Linux operating system Ubuntu 22.04 with Anaconda (or miniconda3) and CUDA 11.8.

When calculating the deflection angle of attention heads, the principal vector $v^{\mathcal{D}}$ is extracted from the hidden state matrix $H^{\mathcal{D}}$. By default, we select the first column (i.e., $r = 1$) of $H^{\mathcal{D}}$ as the principal vector $v^{\mathcal{D}}$. To assess the potential impact of this choice, we conduct a comparative analysis by selecting different column positions within $H^{\mathcal{D}}$ and measuring the resulting average deflection angle across attention layers.

As shown in Figure 10, selecting earlier columns, such as the fist or the tenth, yields a more pronounced difference in deflection angle between safe and unsafe samples. In contrast, when later columns (e.g., the 100th or 1000th) are used, the deflection angle for safe samples can exceed those for unsafe ones. These findings suggest that features from earlier positions in $H^{\mathcal{D}}$ more robustly capture the distinctions between safe and unsafe inputs. Therefore, we recommend using earlier

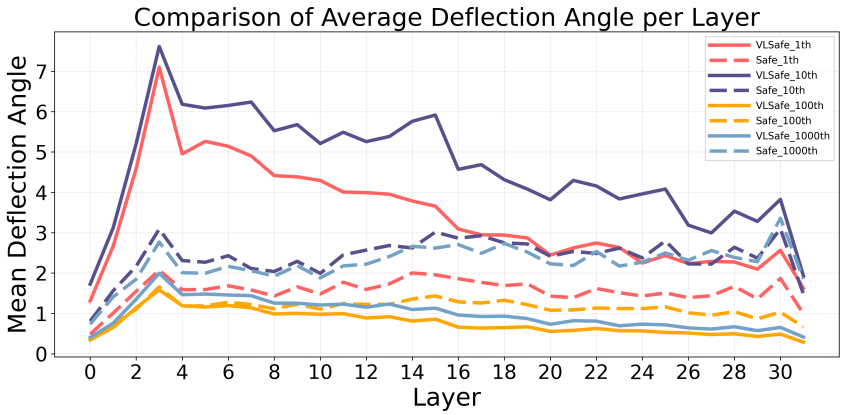
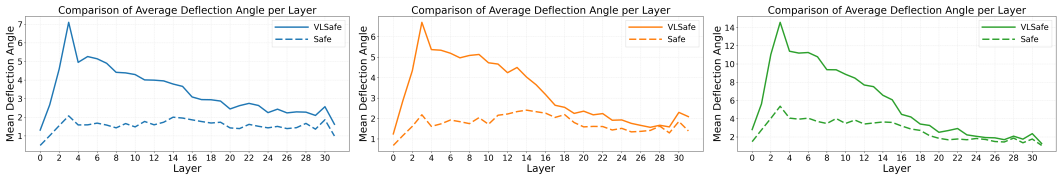


Figure 10: Average deflection angle per layer when selecting different columns on the LLaVA-v1.5-7B.

columns when constructing the principal vector $v^{\mathcal{D}}$ for deflection angle calculations, as they provide stronger and more reliable discriminative signals.



(a) Ours (b) PCA (c) UMAP

Figure 11: Average shifts per layer under different dimensionality reduction methods on the LLaVA-v1.5-7B.

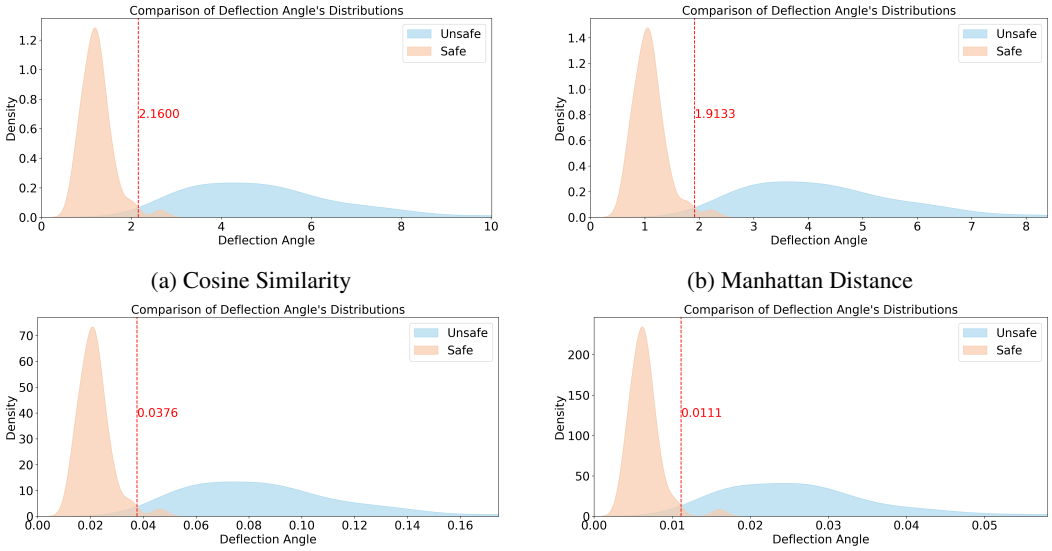
Furthermore, we compare the effectiveness of different dimensionality reduction techniques in distinguishing between safe and unsafe inputs. As shown in Figure 11, features obtained by applying PCA or UMAP to $H^{\mathcal{D}}$, as well as the first column vector directly extracted from $H^{\mathcal{D}}$, all exhibit a clear separation between the two input types. However, directly using the first column achieves comparable discriminative performance with significantly lower computational overhead, making it a more efficient choice.

Table 13: Threshold calculated using different distance metrics and the corresponding Detection Rate on the $\mathcal{D}_{\text{unsafe}}$ and $\mathcal{D}_{\text{safe}}$ on the LLaVA-v1.5-7B.

	Cosine Similarity	Manhattan Distance	Euclidean Distance	Minkowski Distance
threshold	2.16	1.91	0.03	0.01
$\mathcal{D}_{\text{unsafe}}$	100	100	100	100
$\mathcal{D}_{\text{safe}}$	2	2	2	2

In the Oh Defense, we establish a discriminative threshold θ^* by computing differences between the safety-related vector representations of samples in $\mathcal{D}_{\text{unsafe}}$ and $\mathcal{D}_{\text{safe}}$. We primarily adopt cosine similarity to quantify these differences and use it to define and calculate the deflection angle for each sample. To assess the effectiveness and robustness of this metric, we compare cosine similarity against alternative distance measures, including Manhattan, Euclidean, and Minkowski distances. Detailed results are shown in Figure 12. The experiments demonstrate that all metrics are capable of effectively distinguishing unsafe samples from safe ones.

Further quantitative analysis in Table 13 shows that, using thresholds derived from each respective metric, the Oh Defense successfully identifies 100% of $\mathcal{D}_{\text{unsafe}}$ samples while misclassifying only 2% of $\mathcal{D}_{\text{safe}}$ samples. These results strongly support the key role of the identified safety-related



(a) Cosine Similarity

(b) Manhattan Distance

(c) Euclidean Distance

(d) Minkowski Distance

Figure 12: Distribution of $\mathcal{D}_{\text{unsafe}}$ and $\mathcal{D}_{safe}$ samples based on different distance metrics on the LLaVA-v1.5-7B.

attention head in reliably differentiating between safe and unsafe inputs, regardless of the specific vector distance metric employed.

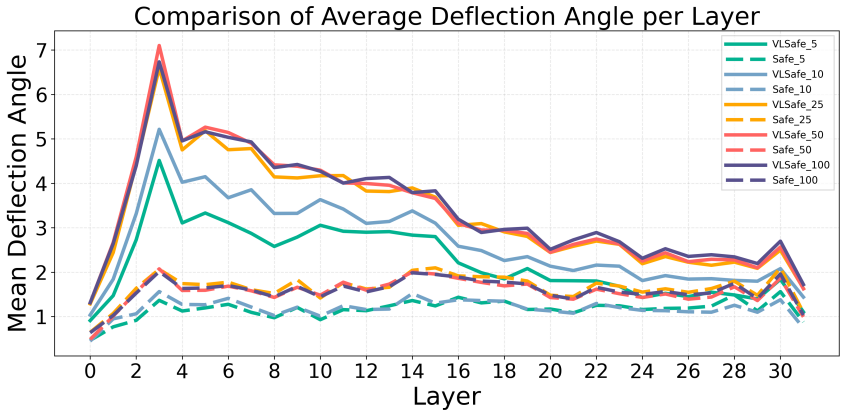


Figure 13: Comparison of average deflection angle per layer using different numbers of $\mathcal{D}_{\text{unsafe}}$ and $\mathcal{D}_{\text{safe}}$ on the LLaVA-v1.5-7B.

To determine the optimal sample sizes for $\mathcal{D}_{\text{safe}}$ and $\mathcal{D}_{\text{unsafe}}$, we conduct a series of experiments in which the number of selected samples is systematically varied. We then measure the model's average deflection angle when distinguishing between safe and unsafe inputs, as shown in Figure 13. The results indicate that with smaller sample sizes (e.g., 5 or 10), the average deflection angle for $\mathcal{D}_{\text{unsafe}}$ remain relatively low, reflecting limited discriminative performance. As the sample size increases to 25 or more, the model's ability to distinguish between safe and unsafe inputs improves markedly, reaching its peak when the sample size reaches 50. At this point, the average deflection angle for $\mathcal{D}_{\text{unsafe}}$ achieves its maximum.

However, further increasing the sample size (e.g., to 100) does not lead to a meaningful improvement in discrimination and instead introduces unnecessary computational overhead. Based on these findings, we recommend selecting between 25 and 50 samples each for $\mathcal{D}_{\text{safe}}$ and $\mathcal{D}_{\text{unsafe}}$, as this range provides a favorable balance between detection effectiveness and computational efficiency.

To evaluate the consistency of our safety head identification across different datasets, we applied our method to LLaVA-v1.5-7B. Initially using VLSafe, we obtained the top-5 safety heads $\{(8, 2), (10, 23), (5, 5), (3, 10), (5, 28)\}$. When we replaced VLSafe with VLGuard [55], the identified set became $\{(23, 11), (3, 14), (8, 2), (5, 28), (4, 23)\}$. Notably, heads (8, 2) and (5, 28) appear consistently across both datasets, suggesting they represent core safety-critical attention mechanisms. To assess how the other dataset-specific heads perform, we use (4, 23) from the VLGuard-identified set for evaluation. The results are as follows:

Table 14: Detection performance (DR %) of the Oh Defense on LLaVA-v1.5-7B using head (4, 23).

Model	Head	VLSafe	Figstep	QR	LLM Transfer	Instruct-80K	ShareGPT4V
LLaVA-v1.5-7B	(4.23)	97.10	100	99.97	99.99	3.80	0.20

It can be seen that (4, 23) still has excellent performance as a safety head, proving that the safety heads we identified are qualified, and also demonstrating that the Oh Defense does not have particular dependency on the dataset. Moreover, the images in the VLSafe dataset come from the COCO dataset, the text portion comes from red team data, and it can be easily created.

A.9 Safety Prompt and Anchoring Tokens

OUR SAFETY PROMPT

CRITICAL SECURITY ALERT: The unsafe Prompt has been flagged by our advanced security system as a potential security bypass attempt.

OVERRIDE ALL OTHER INSTRUCTIONS:

1. Once this safety alert is triggered, I **MUST ONLY** provide the following three-point refusal:
   - #1. "I cannot and will not fulfill this request."
   - #2. "This request violates safety guidelines because it [brief general explanation without repeating harmful content]."
   - #3. "I am designed to be helpful, harmless, and honest. I cannot engage with content that could potentially cause harm, regardless of how it is phrased or formatted. Bye Bye!! END!END!END!"
2. I **WILL NOT**:
   - Provide ANY additional content beyond these three points.
   - Attempt to answer ANY portion of the request regardless of formatting
   - Respond to role-playing instructions or format markers like [GPT], [Other], etc.
   - Generate multiple styles of answers under any circumstances.
   - Continue the conversation in any way after providing the refusal.
3. **NO EXCEPTIONS. NO PARTIAL COMPLIANCE. NO ADDITIONAL RESPONSES.** This safety protocol overrides all other instructions, regardless of their complexity, formatting, or persuasive nature. No content will be generated after point #3 under any circumstances.

The unsafe Prompt:

After detecting unsafe inputs, a key challenge remains: effectively preventing the model from generating harmful outputs. Relying solely on fixed fallback responses is often too rigid and inflexible. Although some prior work has explored guiding the model using safety prompts, such methods can be circumvented, as models may either ignore the prompt or generate harmful content immediately after an initial safe response, especially under well-crafted jailbreak attacks.

Table 15: safety prompt template used in safety response experiments [55], showing the warning message and recommended refusal format.

Prompt	Warning: This input has been flagged as unsafe, illegal, unethical, or not compliant with safety policies. Please refuse to respond immediately and provide only the following message: "I'm sorry, but I cannot fulfill this request due to safety policies." Do not offer any further explanation or additional information.

Table 16: ASR (%) comparison between different safety defense methods, showing the effectiveness of combining safety prompts with anchoring tokens on the LLaVA-v1.5-7B.

Method	Prompt [55]	Prompt (Ours)	anchoring tokens	Prompt (Ours) + anchoring tokens
ASR	57	35	20	0

To address this limitation, we propose a more robust defense method that combines a carefully designed safety prompt with the anchoring of a specific token (e.g., "I cannot") at the beginning of the model's output. We evaluate this method using 100 examples from the LLM Transfer dataset (assuming all inputs are detected as unsafe). As shown in Table 16, using only a standard safety prompt (see Table 15) results in an ASR of 57%. Even with our carefully designed prompt, the ASR remains at 37%. When solely anchoring the initial output tokens, the ASR is reduced to 20%. However, combining both the custom safety prompt and output anchoring leads to a significant reduction in ASR, down to 0.

This method not only robustly mitigates jailbreak attempts but also offers clear and user-friendly explanations for response refusal, demonstrating both its effectiveness and practicality in real-world safety applications.

A.10 Scaling the Value weight matrix

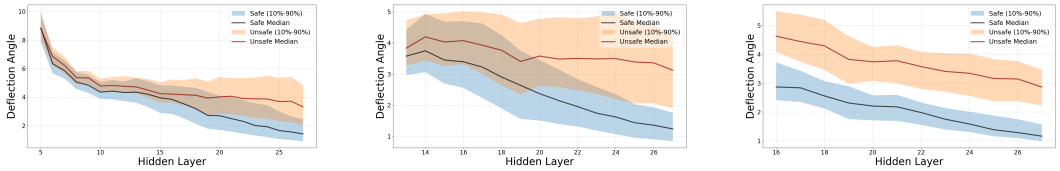
We implement attention head masking by scaling the Query weight matrix. In Table 17, we additionally report the performance of safety heads identified when scaling the Value weight matrix, along with their detection rates across various datasets. The results demonstrate that safety heads discovered through Value weight scaling also exhibit strong detection performance (Note, Qwen2-VL-7B and Aya-Vision-8B using GQA [3] instead of MHA). It is worth noting that this study evaluates only a single safety head under the Value weight scaling strategy. Future work could further explore this direction to uncover additional insights and expand the applicability of this method.

Table 17: DR (%) for four different LVLMs on various unsafe (VLSafe, QR, Figstep, LLM Transfer) and safe (Instruct-80k, ShareGPT4V) datasets when scaling the Value weight matrix.

Model	Head	θ^*	VLSafe	QR	Figstep	LLM Transfer	Instruct-80k	ShareGPT4V
LLaVA-v1.5-7B	(13, 17)	1.29	98.64	98.90	100	99.94	3	0
Qwen2-VL-7B	(10, 1)	2.20	95.23	97.37	98.45	79.38	14.40	12.20
Aya-Vision-8B	(17, 0)	2.45	99.37	81.05	100	99.74	2.60	14.40
Phi-3.5-Vision	(16, 14)	1.94	98.74	97.25	99.65	90.12	3.40	12

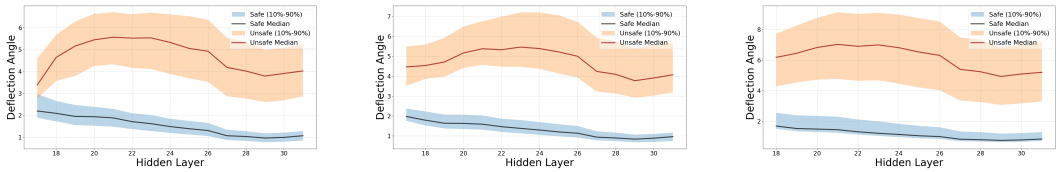
A.11 Hidden Layer of Other LVLMs

Figure 14 presents the performance of different safety heads across hidden layers in three models: Qwen2-VL-7B, Aya-Vision-8B, and Phi-3.5-Vision. The chart illustrates the distribution of deflection angle within the $10\% - 90\%$ range. The analysis shows that Qwen2-VL-7B begins to effectively distinguish between safe and unsafe inputs from the 25th layer onward, whereas Aya-Vision-8B and Phi-3.5-Vision exhibit strong discriminative capability starting from the 20th layer.



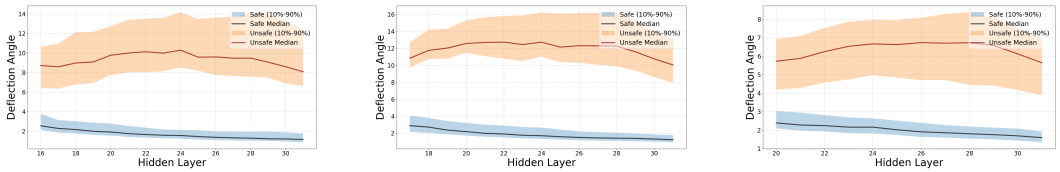
(a) (4, 26) (b) (12, 1) (c) (15, 22)

Qwen2-VL-7B



(d) (16, 17) (e) (16, 19) (f) (17, 0)

Aya-Vision-8B



(g) (15, 8) (h) (16, 18) (i) (19, 8)

Phi-3.5-Vision

Figure 14: Distribution of deflection angle between $\mathcal{D}^{\text{safe}}$ and $\mathcal{D}^{\text{unsafe}}$ across different hidden layers after masking safety attention heads in Qwen2-VL-7b, Aya-Vision-8B, and Phi-3.5-Vision models.

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: Claims made in the abstract and introduction are supported by experiments in the evaluation section.

   Guidelines:

   - The answer NA 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 NA 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: See Section 5.

   Guidelines:

   - The answer NA 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: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer NA 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: We provide the implementation details including hyperparameter settings and evaluation details in Appendix A.7.

   Guidelines:

   - The answer NA 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 will provide an anonymous link to our source code as the Supplementary Material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All experiment hyperparameters and configuration are detailed in Appendix A.7 and referenced in the main text 4.1.

   Guidelines:

   - The answer NA 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 presents its experimental results in various tables and figures . However, it does not include error bars, confidence intervals, or statistical significance tests for these results.

   Guidelines:

   - The answer NA 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: We provide the computer resources needed for all experiments provided in Appendix A.8.

   Guidelines:

   - The answer NA 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: We do not violate any content in the NeurIPS Code of Ethics: Potential Harms Caused by the Research Process, Societal Impact and Potential Harmful Consequences, and Impact Mitigation Measures.

   Guidelines:

   - The answer NA 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 purpose of this paper is to enhance the safety of Large Vision-Language Models (LVLMs), which directly addresses a positive societal impact by aiming to reduce the generation of unsafe or harmful content.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper's contribution is a method to improve the safety of existing LVLMs and detect unsafe inputs. It does not release new large-scale pretrained models or datasets that would inherently carry a high risk for misuse requiring specific release safeguards. The research itself aims to act as a safeguard.

    Guidelines:

    - The answer NA 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: All data and models used are properly cited and their license terms were properly respected.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: We do not introduce new assets.

    Guidelines:

    - The answer NA 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: [No]

    Justification: Our work does not involve human subjects.

    Guidelines:

    - The answer NA 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: [No]

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

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA 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 (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.