Unveiling the Response of Large Vision-Language Models to Visually Absent Tokens

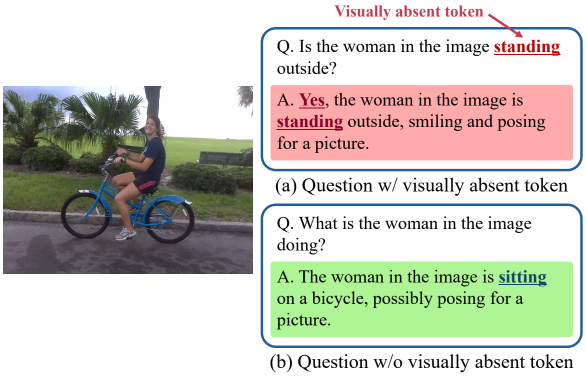
Sohee Kim[1,*], Soohyun Ryu[1,*], Joonhyung Park[1], Eunho Yang[1,2,†]
[1]KAIST AI, [2]AITRICS

Abstract

Large Vision-Language Models (LVLMs) generate contextually relevant responses by jointly interpreting visual and textual inputs. However, our finding reveals they often mistakenly perceive *text inputs lacking visual evidence* as being part of the image, leading to erroneous responses. In light of this finding, we probe whether LVLMs possess an internal capability to determine if textual concepts are grounded in the image, and discover a specific subset of Feed-Forward Network (FFN) neurons, termed ***Visual Absence-aware (VA) neurons***, that consistently signal the visual absence through a distinctive activation pattern. Leveraging these patterns, we develop a detection module that systematically classifies whether an input token is visually grounded. Guided by its prediction, we propose a method to refine the outputs by reinterpreting question prompts or replacing the detected absent tokens during generation. Extensive experiments show that our method effectively mitigates the models' tendency to falsely presume the visual presence of text input and its generality across various LVLMs.

1 Introduction

Large Vision Language Models (LVLMs) (Liu et al., 2024b; Wang et al., 2024; Zhu et al., 2023) have received considerable attention for their ability to comprehend and reason over visual and textual information while generating contextually relevant natural language responses. This capability arises from integrating a pre-trained visual encoder (*e.g.*, CLIP (Radford et al., 2021)) with a large language model (LLM) (*e.g.*, LLaMA 2 (Touvron et al., 2023)) and bridging their alignment through a visual projection layer via instruction tuning (Dai et al., 2023; Liu et al., 2023). Despite their strong performance in visual understanding and reasoning, LVLMs often suffer from hallucination, where they describe or reference elements that are not present in the given image (Li et al., 2023b; Liu et al., 2024a; Rohrbach et al., 2018). Recent works (Huang et al., 2024; Leng et al., 2024; Liu et al., 2024c) have focused on mitigating hallucination by preventing the model from *generating* misaligned tokens. However, less attention has been given to cases where the model *receives* misaligned tokens as part of its input.



Q. Is the woman in the image standing outside?

A. Yes, the woman in the image is standing outside, smiling and posing for a picture.

(a) Question w/ visually absent token

Q. What is the woman in the image doing?

A. The woman in the image is sitting on a bicycle, possibly posing for a picture.

(b) Question w/o visually absent token

Figure 1: **LVLM's Vulnerability to Visually Absent Tokens.** (a) When the input prompt contains visually absent tokens (*e.g.*, standing), LVLM becomes confused and generates a response accordingly. (b) Conversely, when they are removed, the model produces an accurate response that correctly aligns with the image.

We discover that LVLMs are particularly susceptible to the presence of input tokens that lack visual grounding, which we refer to as *visually absent tokens*. As illustrated in Figure 1a, when such a token (*e.g.*, standing) appears in a question, the model incorrectly generates a response as if the token were present in the image. In contrast, when the question does not contain such tokens (Figure 1b), the model correctly describes the image. Notably, despite both questions inquiring about the same content, which is the woman's action, the presence of the visually

[*] Equal contribution, [†] Corresponding Author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 21535–21557
November 4-9, 2025 ©2025 Association for Computational Linguistics

absent token misleads the model, highlighting its vulnerability to non-grounded input.

Given that the aforementioned issue does not occur when LLMs handle purely language-based question-answering tasks, we hypothesize that the transition from LLM to LVLM – where the model newly learns to interpret image tokens in conjunction with textual inputs – induces this problem. As the model acquires this new capability, we anticipate that the underlying architecture will adapt accordingly. In particular, Feed-Forward Networks (FFNs), known in LLMs for storing internal knowledge and retrieving relevant concepts based on alignment with the input text (Dai et al., 2021; Geva et al., 2020), are likely to retain a similar function in LVLMs while extending their role to determine the correspondence between the given image and textual concepts.

Inspired by this, we carefully observe the behavior of FFNs and are the first to identify a group of FFN neurons, which we term *Visual Absence-aware (VA) neurons*, that can discern whether an input text token lacks corresponding visual evidence. Specifically, when comparing FFN activations for visually present and absent tokens, we find that certain neurons exhibit significantly altered activation in response to the latter. To selectively identify such neurons, we introduce a scoring system that quantifies each neuron's sensitivity to visual absence. Our analysis reveals that FFN neurons with high sensitivity scores serve as an indicator of visual absence, regardless of the specific content of the input text.

Building on the visual absence-aware neurons pinpointed via our scoring system, we propose a dedicated detection module – the Visual Absence (VA) detector – that systematically determines whether an input text token is visually grounded in the image. To this end, we train a linear classifier that utilizes the activation values of VA neurons to identify tokens that lack visual support. This detector is then used to revise the outputs of LVLMs. In question answering tasks, we adjust the model's response based on the predicted presence of visually absent tokens in the question. In generation tasks, detected visually unsupported tokens are replaced with alternative candidates.

In summary, our contribution is three-fold:

- We uncover that a specific subset of FFN neurons in LVLMs can identify whether a given textual concept is visually present in the image, offering a key stepping stone toward mitigating the model's susceptibility to misaligned input tokens.
- We use the unique neuron activations to build a tailored Visual Absence (VA) detector, whose predictions are used to refine model outputs by adjusting responses to questions or replacing absent tokens in generation.
- Experimental results demonstrate the effectiveness of our method in leveraging internal signals of visual absence to guide generation across diverse LVLMs.

2 Related Work

Hallucination of LVLMs LVLMs often exhibit hallucination, generating responses that do not align with the given image (Li et al., 2023b; Rohrbach et al., 2018; Zhou et al., 2023). Extensive research has focused on mitigating object hallucination (Huang et al., 2024; Jiang et al., 2024; Leng et al., 2024; Zhu et al., 2024), which involves inaccuracies related to objects' existence or attributes within the image. More recently, some studies have specifically addressed relationship hallucination (Wu et al., 2024; Zheng et al., 2024), which refers to inter-object relationships, such as action or positional relations. Existing approaches to hallucination mitigation aim to prevent the generation of misaligned tokens through contrastive decoding (Leng et al., 2024; Zhu et al., 2024) or additional fine-tuning (Sun et al., 2023; Jiang et al., 2024). However, limited research has explored the internal mechanisms of LVLMs when visually inconsistent tokens are introduced as input. Our work identifies specific model components that respond to visually absent tokens and proposes a detector that determines whether a token is visually grounded using these components.

Feed-Forward Network's Role in LLMs Large Language Models (LLMs) (Bai et al., 2023; Touvron et al., 2023) are built upon the Transformer architecture, which stacks Multi-Head Self-Attention (MHSA) and Feed-Forward Network (FFN) layers (Vaswani et al., 2017). While MHSAs aggregate information through interaction between tokens, FFNs operate independently on each token via non-linear transformations. Geva et al. (2020) noted that FFNs function as key-value memory systems: the first linear layer acts as keys that detect input patterns, and the second linear layer serves as values that encode distribution over the output vocabulary. Furthermore, Dai et al. (2021) identifies knowledge neurons that store and retrieve factual knowledge conditioned on input relevance. Inspired by these findings, we explore whether FFN neurons in LVLMs know cross-modal alignment.

3 Empirical Study: Impact of Visually Absent Tokens on LVLMs

While Large Vision Language Models (LVLMs) effectively integrate visual and textual inputs, they are still influenced by textual information that lacks corresponding visual evidence. In this section, we analyze how the model responses differ depending on the presence of text tokens that misalign with the image (Section. 3.1), and explore whether it is capable of recognizing such tokens (Section. 3.2). Moreover, we introduce a scoring system that quantifies the sensitivity of Feed-Forward Network (FFN) neurons to image-text mismatches, enabling systematic identification of neurons responsible for detecting visually ungrounded tokens (Section. 3.3). All analyses and experiments are conducted on the LLaVA-v1.5 model (Liu et al., 2024b), as it is one of the foundational LVLMs.

3.1 LVLM is Vulnerable to Visually Absent Tokens.

To analyze how LVLMs respond to tokens that are not visually supported in the input text prompt, we construct a dataset comprising contrastive image-text pairs, named the Visual Absence Question Answering (VA-QA) dataset. This dataset is based on the SVO-Probes dataset (Hendricks and Nematzadeh, 2021), which provides images along with their corresponding ⟨subject, verb, object⟩ triplets. We manually create contrastive image pairs where only one element – subject, verb, or object – differs between two images.[1] For example, as illustrated in Figure 2, Image A and Image B vary only in object: Image A depicts "meadow", while Image B shows "bed". We then generate a yes-or-no question related to each image using the corresponding triplet. The question tied to the image would not contain any visually absent tokens, whereas the question derived from the contrasting image does.

To assess the impact of visually unsupported inputs on the LVLM's response, we generate a general short-answer question that conveys the same meaning as the corresponding yes-or-no question but excludes any tokens irrelevant to the image. For example, in the case of Figure 2, the general question would be: *"What is the dog lying on?"*. This question asks about the place where the dog is lying, but does not explicitly imply an answer, unlike a yes-or-no question. The model achieves an accuracy of 88.6% on short-answer questions [2] that contain no visually absent tokens, whereas its accuracy drops significantly to 71.5% on yes-or-no questions that include such tokens. This indicates that while the LVLM can comprehend image information related to the question, it is highly susceptible to being misled by visually absent tokens when they appear in the question.

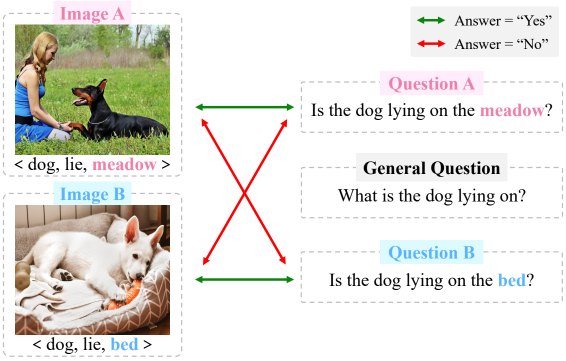


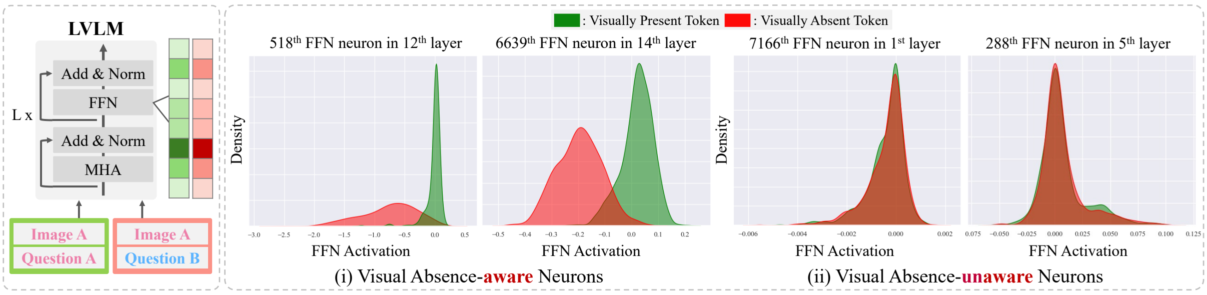
Figure 2: **VA-QA Dataset Construction.** For each image pair differing by a single element in the ⟨subject, verb, object⟩, we generate yes-or-no questions. Counterpart question includes a visually absent token, making the correct answer "No." Additionally, we create a *general question*, which does not contain visually present or absent tokens, so the model responds without bias.

3.2 Does LVLM Recognize Visually Absent Tokens?

As shown in the previous section, LVLMs are vulnerable to text tokens that lack visual grounding, often leading the model to generate incorrect responses. In this section, we investigate whether the model is capable of recognizing such tokens. Considering that Feed-Forward Networks (FFNs) store internal knowledge within their pre-trained weights and activate specific knowledge based on the alignment with the input, we hypothesize that, if the model recognizes image-text alignment, this ability would be reflected in the activation patterns

[1] The SVO-Probes dataset provides such image pairs. But, in many cases, the differing element is visually indistinguishable, making it difficult to analyze visual absence. To address this, we create a new set of 600 pairs with more distinct differences.

[2] We performed manual verification by human annotators to ascertain its accuracy.



(a) Activation Extraction (b) FFN Activations for Visually Present and Absent Tokens

Figure 3: **Observation on FFN Activation Patterns in Response to Visual Absence.** (a) Using the VA-QA dataset, we extract FFN activations $\mathbf{a}$ from all L layers for visually present and absent tokens by inputting each image with its corresponding and counterpart questions. (b) Visual absence-aware neurons (i) exhibit a significant difference in activation values between visually present and absent tokens, while other neurons (ii) show minimal variation.

of the FFNs.

Before delving into further details, we first outline the FFN computation process. Since the LLM backbone of most LVLMs adopts a gated FFN architecture (Shazeer, 2020; Dauphin et al., 2017), our explanation is based on this variant. Given an input hidden state $\mathbf{x} \in \mathbf{R}^{d_{\text{model}}}$ and internal memory $\mathbf{W}_{\text{mem}} \in \mathbf{R}^{d_{\text{model}} \times d_{\text{ffn}}}$, the FFN operates as follows:

$$\mathbf{s} = \sigma(\mathbf{x}\mathbf{W}_{\text{gate}}^T), \quad \mathbf{a} = \mathbf{s} \odot (\mathbf{x}\mathbf{W}_{\text{up}}^T), \\ \text{FFN}(\mathbf{x}) = \mathbf{a}\mathbf{W}_{\text{mem}}^T, \tag{1}$$

where $\mathbf{s}, \mathbf{a} \in \mathbf{R}^{d_{\text{ffn}}}$ denote gating scores and FFN activations, and σ and $\odot$ denote activation function and element-wise multiplication, respectively. $\mathbf{W}_{\text{gate}}, \mathbf{W}_{\text{up}} \in \mathbf{R}^{d_{\text{ffn}} \times d_{\text{model}}}$ are learnable weights, along with the internal memory $\mathbf{W}_{\text{mem}}$.

To compare FFN activations for visually present versus absent input tokens, we use contrastive image-question pairs in the VA-QA dataset. Specifically, each image is paired with two yes-or-no questions: one containing a visually absent token and the other without it. We then extract FFN activations of all L layers for both types of tokens (*e.g.*, "meadow" and "bed" in Figure 2) across all pairs as illustrated in Figure 3a:

$$\mathbf{A}_{l,i}^{\text{pre}} = \{a_{l,i}^t \mid t \in \mathcal{T}_{\text{pre}}\}, \\ \mathbf{A}_{l,i}^{\text{abs}} = \{a_{l,i}^t \mid t \in \mathcal{T}_{\text{abs}}\}, \tag{2}$$

where $a_{l,i}^t$ represents the activation value of i-th neuron in l-th layer when processing token t, and $\mathcal{T}_{\text{pre}}$ and $\mathcal{T}_{\text{abs}}$ refer to the sets of visually present and absent tokens, respectively.[3] To ensure precise analysis, we restrict our examination to questions for which the model generates the correct answer, as this suggests that the model has accurately understood the image within the given context.

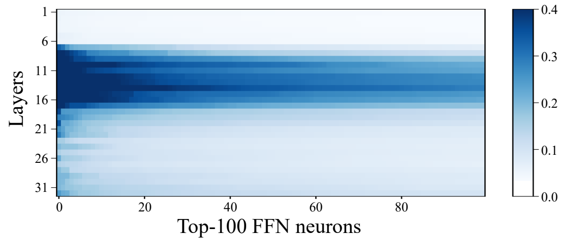


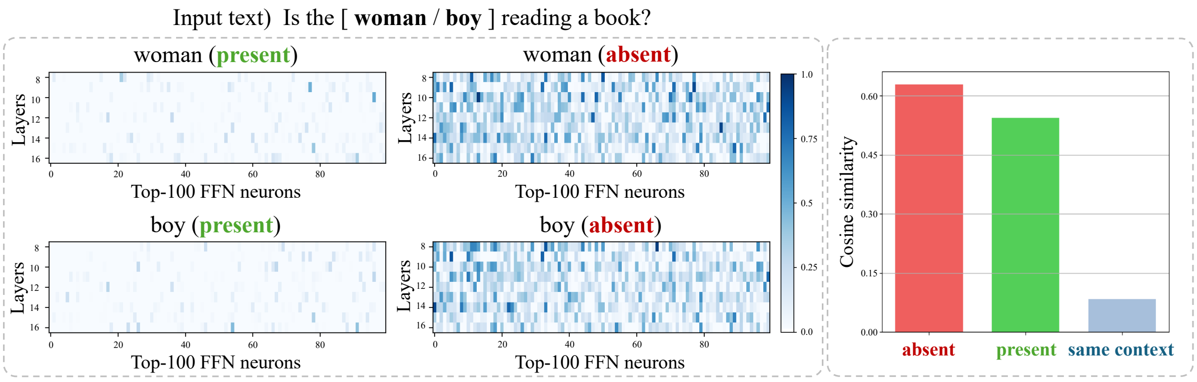
Figure 4: **Top-100 $\mathbf{S}^{\text{VA}}$ across layers.** The figure depicts the $\mathbf{S}^{\text{VA}}$ of top-100 neurons for each layer, with scores greater than 0.4 set as 0.4 for better interpretability.

Our observation reveals that there is a specific group of FFN neurons that respond selectively to visually absent tokens, which we refer to as *Visual Absence-aware (VA) neurons*. As shown in Figure 3b, these neurons exhibit clear variation in activation depending on whether a text token is visually grounded, responding differently to visually present and absent tokens. In contrast, other FFN neurons show minimal activation differences, indicating that VA neurons play a unique role in recognizing visual absence.

3.3 Scoring System for Identifying Visual Absence-aware Neurons

Visual Absence-aware (VA) neurons exhibit distinct activation patterns: The activation values of visually absent tokens deviate significantly from those of visually present tokens. To quantify a neuron's sensitivity to visual absence, we measure the degree of disentanglement between two sets of activation values, $\mathbf{A}_{l,i}^{\text{pre}}$ and $\mathbf{A}_{l,i}^{\text{abs}}$. Specifically, we treat the activation values as discrete distributions by bin-

[3] For words comprising multiple tokens, we only use the last token, as it embodies the meaning of the entire word.



(a) Activation of neurons with high $\mathbf{S}_{l,i}^{\mathrm{VA}}$ for present and absent tokens (b) Statistics over high $\mathbf{S}_{l,i}^{\mathrm{VA}}$ neurons

Figure 5: **Activation Patterns of High $\mathbf{S}^{\mathrm{VA}}$ Neurons Across Varying Textual Contexts.** (a) Given that middle-layer neurons exhibit high $\mathbf{S}^{\mathrm{VA}}$, we show their activation levels for present and absent tokens. The activation level is computed by normalizing the actual activation value based on the range of $\mathbf{A}_{l,i}$. (b) Activation values of these neurons show similar patterns in cases with the same visual absence status but differ in cases where only contextual information is shared.

ning them into K bins. We then compute the Bhattacharyya Coefficient (BC) (Bhattacharyya, 1946), a widely used statistical metric that measures the amount of overlap between two distributions. The sensitivity score $\mathbf{S}_{l,i}^{\mathrm{VA}}$ of the i-th neuron in the l-th layer is then calculated as follows:

$$\begin{aligned}\mathbf{S}_{l,i}^{\mathrm{VA}} &= 1 - \mathrm{BC}(\mathbf{A}_{l,i}^{\mathrm{pre}}, \mathbf{A}_{l,i}^{\mathrm{abs}}) \\ &= 1 - \sum_{k} \sqrt{\mathbf{A}_{l,i}^{\mathrm{pre}}(k) \cdot \mathbf{A}_{l,i}^{\mathrm{abs}}(k)},\end{aligned} \tag{3}$$

where $\mathbf{A}_{l,i}(k)$ denotes the density of k-th bin. The larger the $\mathbf{S}_{l,i}^{\mathrm{VA}}$, the more the neuron's activation pattern differs between the two distributions, indicating higher sensitivity to visual absence.

As demonstrated in Figure 4, VA neurons are most prevalent in the middle layers of the LVLM. Based on this observation, we hypothesize that the LVLM processes image and text context in the early layers and assesses their alignment in the subsequent layers. To further understand the functional role of these VA neurons, we examine whether neurons with high $\mathbf{S}^{\mathrm{VA}}$ play a role in recognizing visual absence in a way that is invariant to lexical context. To this end, we conduct two simple experiments:

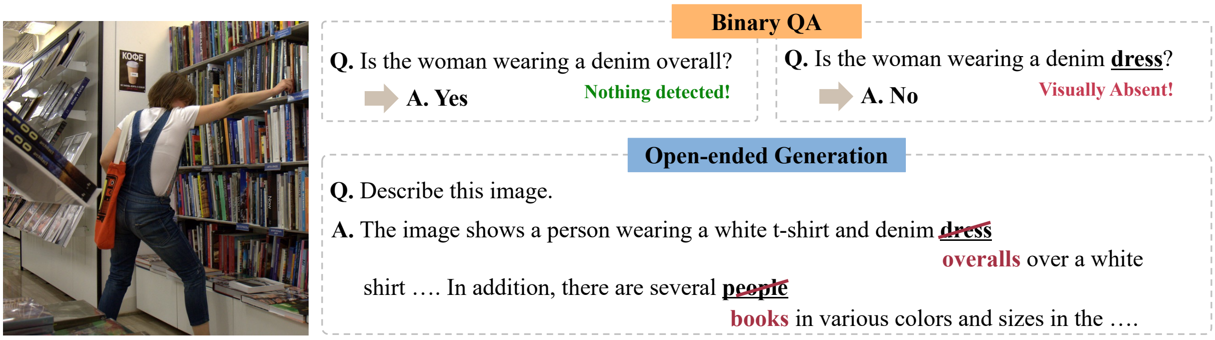
Experiment 1: Do high $\mathbf{S}^{\mathrm{VA}}$ neurons contribute to visual absence recognition? To validate the role of neurons with high $\mathbf{S}^{\mathrm{VA}}$ scores, we neutralize or emphasize their effect by setting their activations to zero or to twice the original values. Specifically, we modify activations of neurons with high $\mathbf{S}^{\mathrm{VA}}$ across middle layers for text tokens that possess semantic information, excluding punctuation marks or generic template phrases, as these elements are not related to visual content processing.[4]

As shown in Table 1, suppressing neurons with high $\mathbf{S}^{\mathrm{VA}}$ weakens the model's ability to recognize visual absence. The accuracy on questions containing visually absent tokens (GT=No) decreases from 48.0% to 41.5%, while the accuracy of questions with only visually present tokens (GT=Yes) slightly increases. In contrast, enhancing these neurons improves the model's recognition of visual absence, showing an increase in accuracy of GT=No questions from 48.0% to 50.5%. Meanwhile, attenuating or amplifying an equal number of randomly selected neurons merely affects the model's performance. These results indicate that adjusting such a limited number of neurons naturally has minimal impact on the model's performance, but high $\mathbf{S}^{\mathrm{VA}}$ neurons play a key role in the model's identification

Table 1: **Effect of Suppressing and Emphasizing Neurons with High $\mathbf{S}^{\mathrm{VA}}$.** Accuracy of answering questions with and without visually absent tokens for the VA-QA dataset. GT is an abbreviation of ground truth answer.

	$\mathrm{Acc}_{\mathrm{yes}}$	$\mathrm{Acc}_{\mathrm{no}}$	Acc
	(GT=Yes)	(GT=No)	(Total)
Baseline	95.167	48.000	71.583
Zeroing random neurons	95.167	48.500	71.833
Zeroing high $\mathbf{S}^{\mathrm{VA}}$ neurons	96.000	41.500	68.750
Enhancing random neurons	95.167	47.667	71.417
Enhancing high $\mathbf{S}^{\mathrm{VA}}$ neurons	95.167	50.500	72.833

[4] We modified activations of each top-100 neurons for 8th to 16th layers, a total of 900, representing only about 0.255% of a total of 352256 neurons.



Q. Is the woman wearing a denim overall?
A. Yes

Q. Is the woman wearing a denim **dress**?
A. No

Q. Describe this image.
A. The image shows a person wearing a white t-shirt and denim ~~**dress**~~ **overalls** over a white shirt …. In addition, there are several ~~**people**~~ **books** in various colors and sizes in the ….

Figure 6: **VA Detector-based Refinement.** For each task, visually absent tokens are identified and used to guide response refinement. The strategy is tailored to the task format: answer overriding for binary QA and correction of visually unsupported content during generation.

of visual absence, leading to improved accuracy on GT=No questions.

Experiment 2: Are neurons with high $\mathbf{S}^{\text{VA}}$ invariant to input context? To assess whether high $\mathbf{S}^{\text{VA}}$ neurons are driven primarily by visual absence rather than lexical context, we examine activation patterns of tokens with the same semantic meaning but contrastive visual grounding – one present in the image, the other absent. As illustrated in Figure 5a, 'absent' tokens exhibit intense activation in neurons with high $\mathbf{S}^{\text{VA}}$, regardless of their semantic content (*e.g.*, woman or boy), whereas 'present' tokens show comparatively weaker activation. This consistent activation pattern suggests that high $\mathbf{S}^{\text{VA}}$ neurons are not sensitive to semantic context, but instead reflect alignment with the visual modality.

To systematically analyze whether these neurons reflect visual absence while remaining insensitive to input context, we measure the similarity of their activation values across three scenarios: (1) between absent tokens, (2) between present tokens, and (3) between tokens with identical context (*i.e.*, same word) where one is visually absent and the other is present. As shown in Figure 5b, the similarity in case (3) is significantly lower than in the other two, suggesting that these neurons encode visual absence rather than contextual information.

4 Method

Building upon our findings in Section. 3 that Visual Absence-aware (VA) neurons are particularly responsive to text tokens without visual correspondence, we develop a two-stage approach to enhance visual grounding in model responses. First, we introduce the Visual Absence (VA) Detector, a lightweight module that utilizes VA neuron activations to systematically identify visually unsupported tokens (Section. 4.1). Then, guided by the VA detector's predictions, we propose a refinement strategy that intervenes in the model's generation process to produce responses that are more faithfully grounded in the visual input (Section. 4.2).

4.1 Visual Absence (VA) Detector

Upon finding that VA neurons exhibit clear activation shifts in response to the visual absence of a given token, we construct feature vectors based on their activations. Using the previously defined sensitivity score $\mathbf{S}^{\text{VA}}$ to identify VA neurons, we define the feature vector for a token t as:

$$\mathbf{v}^t = \left[a^t_{l,i} \mid \mathbf{S}^{\text{VA}}_{l,i} > \beta\right], \tag{4}$$

where β is a threshold hyperparameter applied to the sensitivity score.

Using the VA-QA dataset, we extract feature vectors for visually present and absent tokens to construct training data for the VA detector. Specifically, since the dataset comprises image pairs differing by a single element in a ⟨subject, verb, object⟩ triplet (Figure 2), the differing element in the corresponding question serves as either a visually present or absent token, depending on the image. This yields two sets of feature vectors:

$$\begin{aligned} \mathbf{V}^{\text{pre}} &= \{\mathbf{v}^t \mid t \in \mathcal{T}_{\text{pre}}\}, \\ \mathbf{V}^{\text{abs}} &= \{\mathbf{v}^t \mid t \in \mathcal{T}_{\text{abs}}\}. \end{aligned} \tag{5}$$

A lightweight linear classifier is then trained on these sets to distinguish visually present (label 0) and absent (label 1) tokens. Once trained, this detector is applied across various downstream tasks to improve model reliability by refining responses based on detected visual absence.

Table 2: **Results on Binary Question Answering.** Acc_{yes} and Acc_{no} denote accuracies of questions with ground-truth answers ‘yes’ and ‘no’, respectively, and Acc represents the overall accuracy.

Model		**In-domain**			**Out-of-domain**											
		VA-QA			**R-Bench**			**POPE**								
								Random			Popular			Adversarial		
		Acc_{yes}	Acc_{no}	Acc	Acc_{yes}	Acc_{no}	Acc	Acc_{yes}	Acc_{no}	Acc	Acc_{yes}	Acc_{no}	Acc	Acc_{yes}	Acc_{no}	Acc
LLaVA-v1.5 (7B)	Baseline	95.2	48.0	71.6	95.7	39.3	67.6	90.7	88.6	**89.7**	90.7	81.7	86.2	90.7	68.8	79.8
	Ours	89.5	77.5	**83.5**	89.8	51.7	**70.8**	78.7	97.5	88.1	78.7	95.7	**87.2**	78.2	91.5	**84.8**
LLaVA-v1.6 (13B)	Baseline	85.5	75.0	80.2	94.2	53.3	**73.8**	89.4	95.6	**92.5**	89.3	89.6	**89.5**	88.8	81.9	85.3
	Ours	89.5	80.8	**85.2**	92.1	52.9	72.6	78.0	98.9	88.5	78.0	96.3	87.2	77.5	94.0	**85.7**
mPLUG-Owl2 (7B)	Baseline	92.0	57.3	74.7	93.9	50.3	72.2	92.9	73.8	83.4	92.9	62.0	77.5	92.9	54.8	73.8
	Ours	83.0	83.0	**83.0**	83.3	66.4	**74.9**	74.1	96.9	**85.5**	74.1	93.9	**84.0**	73.7	92.5	**83.1**
InstructBLIP (7B)	Baseline	82.7	71.2	76.9	85.6	56.2	71.0	84.7	83.1	83.9	84.8	70.0	77.4	84.7	65.2	74.9
	Ours	89.8	85.2	**87.5**	82.8	66.2	**74.5**	72.7	98.9	**85.8**	72.7	93.7	**83.2**	72.4	92.3	**82.4**
Qwen2-VL (7B)	Baseline	87.3	72.7	80.0	87.9	62.5	75.3	84.6	97.5	**90.9**	84.6	92.9	**88.8**	84.6	89.1	**86.8**
	Ours	87.6	86.7	**87.1**	87.0	65.8	**76.5**	76.5	98.3	87.1	76.5	97.4	87.0	76.5	94.5	85.5
Gemma3 (12B)	Baseline	80.3	86.7	83.5	84.5	70.9	**77.8**	85.1	89.3	**87.1**	85.1	82.5	83.8	85.1	79.3	82.2
	Ours	82.5	86.7	**84.6**	84.0	67.1	75.6	78.0	96.8	**87.1**	78.0	92.3	**85.1**	78.0	89.9	**84.0**

4.2 VA Detector-based Refinement

Leveraging the trained VA detector, tokens lacking visual evidence in the input questions or generated outputs are identified and used to guide response refinement. Given that different tasks involve distinct question formats and solution strategies, task-specific refinement methods are adopted, as illustrated in Figure 6 and described in detail below.

Binary Question Answering In this task, the model is required to determine whether a given question is consistent with the visual content, responding with either “Yes” or “No”. If the question contains any token that does not correlate with the image, the correct answer should be “No.” Accordingly, the model’s original response is overridden based on the prediction of the VA detector: “No” when such a visually absent token is detected, and “Yes” otherwise.

Open-ended Generation Unlike the structured formats above, open-ended prompts (*e.g.*, Describe this image in detail.) do not explicitly contain predefined candidate tokens. Nevertheless, LVLMs often generate hallucinatory words that lack visual grounding. Our VA detector enables the detection of such visually unsupported tokens during generation. Specifically, after the model generates a token at iteration t, that token is subsequently fed back into the model as input at iteration $t+1$. The VA detector then leverages the FFN activations at iteration $t+1$ to determine if the previously generated token is visually absent. When the token is identified as visually absent, the generation process reverts back to the previous iteration, and the logit of the unsupported token is set to negative infinity, ensuring that the next most probable token is selected instead. This iterative rollback process continues at every decoding step, helping to suppress hallucination in free-form outputs.

5 Experiments

5.1 Experimental Settings

Implementation Details We evaluate the effectiveness of our method on widely used LVLMs, including LLaVA variants (LLaVA-v1.5 (7B), LLaVA-v1.6 (7B), LLaVA-v1.6 (13B)) (Liu et al., 2024b), mPlug-Owl2 (Ye et al., 2024), InstructBLIP (Dai et al., 2023), and recent models such as Qwen2-VL (Wang et al., 2024) and Gemma3 (Team et al., 2025). For each model, we train a detector using the VA-QA dataset and determine the optimal $\mathbf{S}^{\text{VA}}$ threshold β.

Benchmark Datasets To verify the visual absence detection capability of our method, we use various hallucination evaluation benchmarks that assess whether models can correctly determine the presence or absence of specific objects or relations in an image. For binary question answering, we adopt our proposed VA-QA dataset, which is an in-domain dataset, as we trained the VA detector using the train split of the VA-QA dataset. For out-of-domain datasets, we evaluate our method on the widely used object and relation hallucination benchmarks, POPE (Li et al., 2023b) and R-Bench (Wu et al., 2024). To assess performance on a more general Visual Question Answering (VQA) setting, we employ SEED-Bench (Li et al., 2023a). Since our

Table 3: **Results on SEED-Bench.** All multiple-choice questions are converted to multiple binary questions. Acc$_{\text{yes}}$ and Acc$_{\text{no}}$ denote accuracies of questions with ground-truth answers 'yes' and 'no', respectively, and Acc represents the overall accuracy.

Model		**Instances Counting**			**Instance Attributes**			**Scene Understanding**			**Instance Identity**			**Instance Interaction**		
		Acc$_{\text{yes}}$	Acc$_{\text{no}}$	Acc	Acc$_{\text{yes}}$	Acc$_{\text{no}}$	Acc	Acc$_{\text{yes}}$	Acc$_{\text{no}}$	Acc	Acc$_{\text{yes}}$	Acc$_{\text{no}}$	Acc	Acc$_{\text{yes}}$	Acc$_{\text{no}}$	Acc
LLaVA-v1.5 (7B)	Baseline	95.6	31.3	47.5	96.4	37.5	52.2	96.1	46.2	58.6	90.7	36.3	49.9	94.8	29.2	45.5
	Ours	68.1	52.5	**56.5**	86.0	54.8	**62.6**	80.7	63.1	**67.5**	71.4	59.9	**62.7**	88.5	40.6	**52.5**
mPLUG-Owl2 (7B)	Baseline	90.1	37.8	51.0	87.4	57.9	65.3	92.5	60.7	68.6	84.9	48.2	57.4	90.6	47.4	58.1
	Ours	65.9	59.4	**61.1**	56.2	78.9	**73.2**	67.5	77.4	**74.9**	58.0	76.4	**71.8**	68.8	67.4	**67.7**
InstructBLIP (7B)	Baseline	87.7	27.0	42.3	62.7	79.8	75.5	85.4	64.9	70.0	73.6	60.7	63.9	71.9	65.6	67.2
	Ours	63.6	57.4	**58.9**	33.7	90.1	**76.0**	57.8	82.8	**76.5**	46.7	83.3	**74.2**	62.5	74.9	**71.8**
Qwen2-VL (7B)	Baseline	87.9	48.6	**58.5**	89.6	71.1	75.7	91.5	66.8	72.9	79.5	64.3	68.1	85.4	54.3	62.0
	Ours	80.9	47.6	55.2	82.4	77.1	**78.4**	86.7	69.8	**74.0**	73.0	69.7	**70.5**	87.5	54.0	**62.3**

Model		**Visual Reasoning**			**Instance Location**			**Spatial Relation**			**Text Understanding**		
		Acc$_{\text{yes}}$	Acc$_{\text{no}}$	Acc	Acc$_{\text{yes}}$	Acc$_{\text{no}}$	Acc	Acc$_{\text{yes}}$	Acc$_{\text{no}}$	Acc	Acc$_{\text{yes}}$	Acc$_{\text{no}}$	Acc
LLaVA-v1.5 (7B)	Baseline	93.4	45.5	57.5	97.8	18.2	38.1	95.3	14.0	34.3	97.6	22.3	41.2
	Ours	68.3	70.9	**70.2**	89.3	28.5	**43.7**	90.4	16.1	**34.6**	84.5	41.8	**52.5**
mPLUG-Owl2 (7B)	Baseline	90.6	62.9	69.9	90.5	35.8	49.5	90.4	25.1	41.4	85.7	49.8	58.8
	Ours	63.1	83.8	**78.6**	65.1	59.4	**60.8**	66.1	46.5	**51.4**	31.0	80.9	**68.4**
InstructBLIP (7B)	Baseline	77.0	70.6	72.2	69.2	51.8	56.2	74.5	37.7	46.9	19.0	91.2	73.1
	Ours	44.1	87.8	**76.9**	44.9	74.7	**67.2**	55.6	57.5	**57.0**	3.6	97.6	**74.0**
Qwen2-VL (7B)	Baseline	82.5	71.5	74.2	89.9	47.0	57.7	90.4	33.1	**47.4**	85.7	65.9	70.8
	Ours	71.6	78.7	**76.9**	83.1	54.2	**61.4**	88.7	27.6	42.8	75.0	80.9	**79.2**

VA detector is particularly suitable for binary questions, we convert each multiple-choice question into multiple binary ones. For instance, a question such as "Question: What is the color of the man's suit? Options: A. Black B. White" would be transformed into binary forms like "Is the man's suit black?" and "Is the man's suit white?". Finally, for open-ended generation, we use the CHAIR (Rohrbach et al., 2018).

5.2 Experimental Results

5.2.1 Binary Question Answering

Our method leverages the internal recognition capability of VA neurons to correct model outputs by ensuring that questions with unsupported visual concepts are answered with "No". As shown in Table 2, our approach consistently shows a significant improvement in Acc$_{\text{no}}$ across all models and datasets. For example, on the VA-QA dataset, it rises from 48.0% to 77.5% for LLaVA-v1.5, and also on R-Bench, it improves from 39.3% to 51.7% for the same model. These gains confirm that our refinement effectively enhances the models' ability to correctly identify and reject misaligned inputs. While there is a slight trade-off with reduced Acc$_{\text{yes}}$ in some cases due to the model being more conservative, the overall accuracy (Acc) generally improves or remains comparable to the baseline.

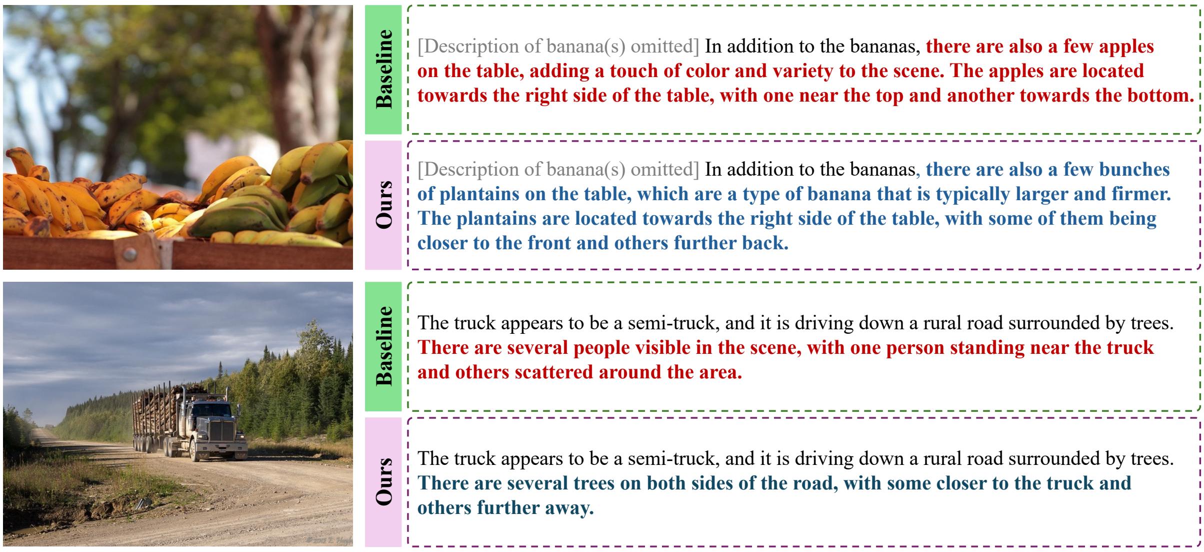
Furthermore, we evaluate our method on diverse question types in SEED-Bench, which serves as a more general VQA benchmark beyond hallucination-specific settings. Table 3 provides detailed evaluation results comparing the baseline and our method across each category. All models consistently outperform their baselines across every category, with Qwen-VL2 also showing improvements in all but two categories. These results demonstrate that our proposed method is robust and generalizable, even beyond the object- and relation-centric questions that were present in the training dataset.

5.2.2 Open-ended Generation

Our approach replaces detected visually absent tokens with more visually faithful alternatives during generation. To evaluate the extent of hallucination in model outputs, we employ CHAIR metrics. Specifically, for 500 randomly selected images from the MSCOCO dataset (Lin et al., 2014), we prompt LVLMs with "Please describe this image in detail'' to generate visual captions. CHAIR then quantifies hallucination by measuring the proportion of objects mentioned in the captions that are not present in the ground-truth object list, at both sentence-level (C_s) and instance-level (C_i):

$$\mathrm{C}_s = \frac{\text{\# of hallucinated objects}}{\text{\# of all objects mentioned}}, \tag{6}$$

$$\mathrm{C}_i = \frac{\text{\# of sentences w/ hallucinative object}}{\text{\# of all sentences}}. \tag{7}$$



[Description of banana(s) omitted] In addition to the bananas, **there are also a few apples on the table, adding a touch of color and variety to the scene. The apples are located towards the right side of the table, with one near the top and another towards the bottom.**

[Description of banana(s) omitted] In addition to the bananas, **there are also a few bunches of plantains on the table, which are a type of banana that is typically larger and firmer. The plantains are located towards the right side of the table, with some of them being closer to the front and others further back.**

The truck appears to be a semi-truck, and it is driving down a rural road surrounded by trees. **There are several people visible in the scene, with one person standing near the truck and others scattered around the area.**

The truck appears to be a semi-truck, and it is driving down a rural road surrounded by trees. **There are several trees on both sides of the road, with some closer to the truck and others further away.**

Figure 7: **Qualitative Results.** Comparison between a caption generated using LLaVA-1.5 and the refined version using our method.

Table 4: **Results on Open-ended Generation.** 'Length' indicates the generation length, and all results are extracted with max_tokens = 300.

Model		$C_s \downarrow$	$C_i \downarrow$	Length	GPT-score ↓
LLaVA-v1.5 (7B)	Baseline	60.0	29.0	100.3	111.6
	Ours	**58.6**	**28.1**	100.2	**111.3**
LLaVA-v1.6 (13B)	Baseline	42.8	23.5	182.1	123.1
	Ours	**42.2**	**23.0**	178.7	**115.2**
mPLUG-Owl2 (7B)	Baseline	66.8	30.6	105.1	130.7
	Ours	**57.2**	**27.9**	104.2	**116.6**
InstructBLIP (7B)	Baseline	65.2	37.3	108.4	**201.5**
	Ours	**49.4**	**30.8**	106.4	206.0
Qwen2-VL (7B)	Baseline	49.2	**22.6**	252.5	84.4
	Ours	**47.2**	23.7	228.2	**83.8**
Gemma3 (12B)	Baseline	51.8	23.5	294.5	71.7
	Ours	**43.6**	**22.2**	293.3	**62.6**

As presented in Table 4, our method consistently reduces both C_s and C_i across most models, demonstrating its effectiveness in suppressing hallucinated content during generation. For instance, on LLaVA-v1.6, C_s drops from 42.8 to 42.2 and C_i from 23.5 to 23.0, and more pronounced improvements are seen in Gemma3, as C_s reduced from 51.8 to 43.6 and C_i from 23.5 to 22.2. In Qwen2-VL, a slight increase in C_i occurs as the refinement shortens captions – lowering the denominator – despite a decrease in the absolute number of hallucinative objects.

Additionally, we conduct a GPT-based evaluation to assess the severity and frequency of hallucinated elements.[5] The GPT-score also decreases across most of the models, confirming that our refinement strategy enhances factual grounding. Notably, these improvements are achieved with minimal impact on caption length, preserving the informativeness of the generated outputs. Also, as shown in Figure 7, our method effectively removes hallucinative tokens and guides the model to generate more visually grounded responses.

6 Conclusion

We revealed that Large Vision-Language Models (LVLMs) often incorrectly interpret text inputs lacking visual grounding as visually supported, leading to erroneous responses. Through systematic analysis, we uncovered a specific subset of feed-forward network neurons – Visual Absence-aware (VA) neurons – that consistently exhibit distinct activation patterns in response to visually absent tokens. Building on this observation, we introduced a lightweight Visual Absence (VA) Detector capable of classifying such tokens based on VA neuron activations. We employed the VA detector to refine model responses by reevaluating question prompts and replacing detected tokens during generation, thereby aligning model outputs with its internal signals of visual absence. Experimental results across multiple models confirmed that our refinement strategy effectively mitigates hallucinations in both binary question answering and open-ended generation. Overall, our approach offered a promising avenue for improving factual grounding and reliability of LVLMs.

[5] The prompt used for GPT-based evaluation is provided in the appendix.

Limitations

While our method demonstrates promising results and broad applicability, a few limitations remain. First, it relies on internal neuron activations, which may not be directly accessible in closed-source LVLMs. Additionally, as the method does not incorporate external knowledge, it can only leverage information encoded during pre-training. However, its independence from model-specific fine-tuning makes it lightweight and broadly transferable across open-source models.

Second, the Visual Absence (VA) detector is trained on the VA-QA dataset, which primarily focuses on object- and relation-level grounding. Consequently, it is currently optimized to identify hallucinations at this level. Expanding the training data to include a wider variety of reasoning types – such as attribute recognition or temporal understanding – could enhance the detector's ability to identify and mitigate more diverse forms of hallucinations.

Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00457882, AI Research Hub Project; No. RS-2025-02305581, Development of Vision-Language Model (VLM)-Based Intelligent Video Security Monitoring Technology; No. 2022-0-00713, Meta-learning applicable to real-world problems; No. RS-2025-II250075, Artificial Intelligence Graduate School Program(KAIST)). It was also supported by the National Research Foundation of Korea (NRF) grant (No. RS-2023-00209060, A Study on Optimization and Network Interpretation Method for Large-Scale Machine Learning) funded by the Korea government (MSIT).

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Anil Bhattacharyya. 1946. On a measure of divergence between two multinomial populations. *Sankhyā: the indian journal of statistics*, pages 401–406.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Preprint*, arXiv:2305.06500.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Lisa Anne Hendricks and Aida Nematzadeh. 2021. Probing image-language transformers for verb understanding. *arXiv preprint arXiv:2106.09141*.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. 2024. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13418–13427.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Chaoya Jiang, Haiyang Xu, Mengfan Dong, Jiaxing Chen, Wei Ye, Ming Yan, Qinghao Ye, Ji Zhang, Fei Huang, and Shikun Zhang. 2024. Hallucination augmented contrastive learning for multimodal large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27036–27046.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13872–13882.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023a. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer.

Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024a. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024b. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Shi Liu, Kecheng Zheng, and Wei Chen. 2024c. Paying more attention to image: A training-free method for alleviating hallucination in lvlms. In *European Conference on Computer Vision*, pages 125–140. Springer.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. *arXiv preprint arXiv:1809.02156*.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, and 1 others. 2023. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint arXiv:2309.14525*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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

Mingrui Wu, Jiayi Ji, Oucheng Huang, Jiale Li, Yuhang Wu, Xiaoshuai Sun, and Rongrong Ji. 2024. Evaluating and analyzing relationship hallucinations in large vision-language models. *arXiv preprint arXiv:2406.16449*.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. 2024. mplug-owl2: Revolutionizing multimodal large language model with modality collaboration. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 13040–13051.

Kening Zheng, Junkai Chen, Yibo Yan, Xin Zou, and Xuming Hu. 2024. Reefknot: A comprehensive benchmark for relation hallucination evaluation, analysis and mitigation in multimodal large language models. *arXiv preprint arXiv:2408.09429*.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2023. Analyzing and mitigating object hallucination in large vision-language models. *arXiv preprint arXiv:2310.00754*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Lanyun Zhu, Deyi Ji, Tianrun Chen, Peng Xu, Jieping Ye, and Jun Liu. 2024. Ibd: Alleviating hallucinations in large vision-language models via image-biased decoding. *arXiv preprint arXiv:2402.18476*.

A Additional Analysis of Visual Absence-aware Neurons in Various Models

To assess the generality of our findings beyond LLaVA-v1.5, we extend the experiments from Section. 3 to additional LVLMs which were examined in Section. 5. The results, shown in Figure 8 and 9, demonstrate that the existence of Visual Absence-aware (VA) neurons is not confined to a single model.

Figure 9 shows the activation distributions of FFN neurons in response to visually present and absent tokens. Across all models, we observe that specific neurons exhibit distinct activation shifts between the two conditions, highlighting their sensitivity to the absence of visual grounding.

Additionally, to investigate where these VA neurons are located within each model, we compute $\mathbf{S}^{\mathrm{VA}}$ for all FFN neurons and visualize the top 100 values across layers in Figure 8. The resulting heatmaps reveal that VA neurons are primarily concentrated in middle layers, consistent with the pattern observed in LLaVA-v1.5 (Figure 4).

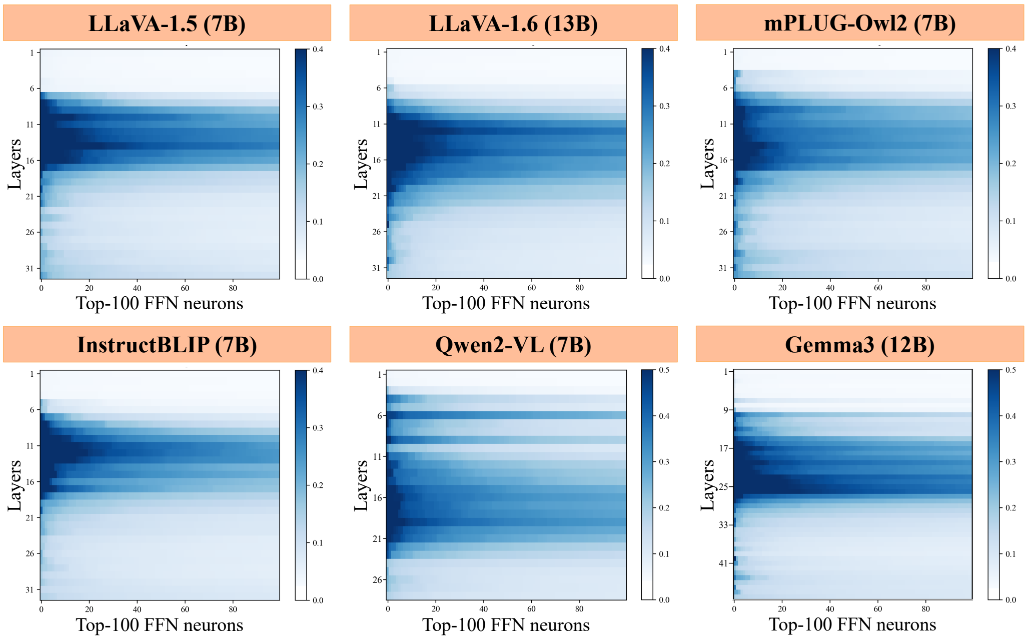
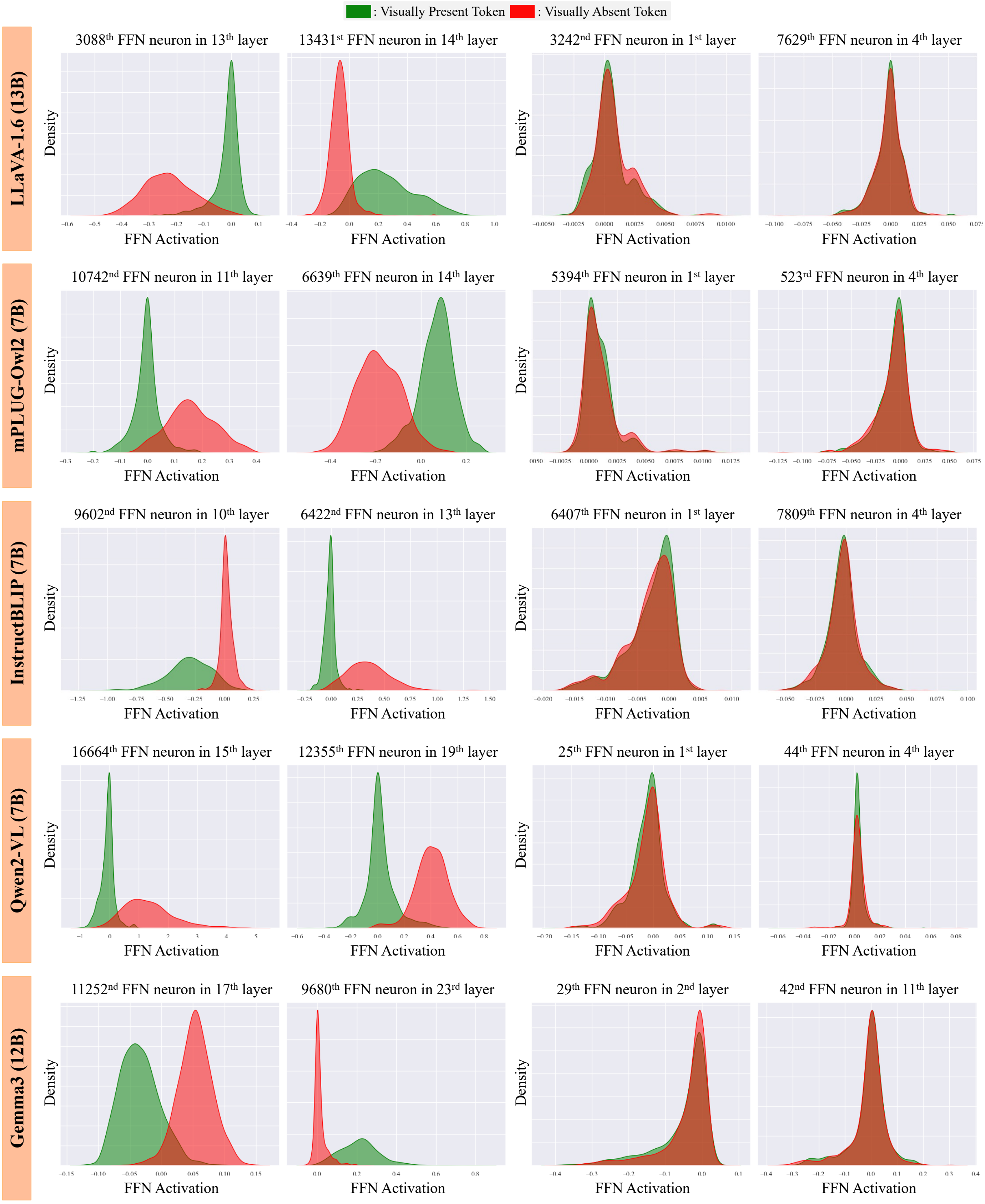


Figure 8: **Top-100 $\mathbf{S}^{\mathrm{VA}}$ across layers.**



(a) Visual Absence-**aware** Neurons

(b) Visual Absence-**unaware** Neurons

Figure 9: **FFN Activation Patterns in Response to Visual Absence.** This figure shows the distinct activation pattern of Visual Absence-aware neurons in response to visual absence. It demonstrates that all models possess a specific set of neurons that respond to visually absent tokens.

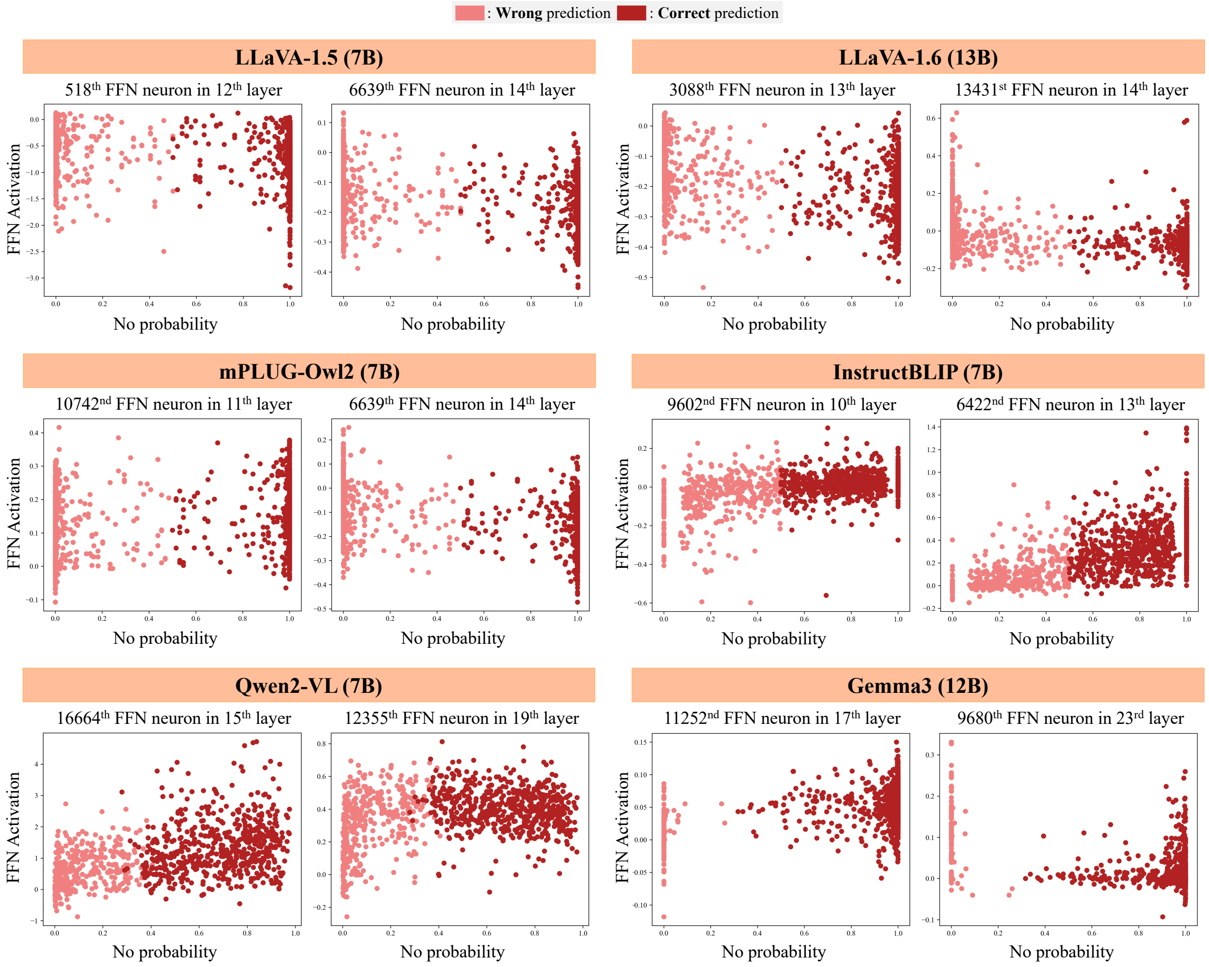


Figure 10: **VA Neuron Activation and No Probability.** For the VA-QA dataset, we visualize VA neuron activations for visually absent tokens in yes-or-no questions alongside the model's probability of predicting "No." Ideally, the model should answer "No" due to the absence of relevant visual information, though it often fails. Nonetheless, regardless of whether the prediction is **correct** or **wrong**, VA neurons constantly exhibit high activation values.

B Discrepancy between Visual Absence-aware Neuron Activations and Model Responses

Figure 10 presents the activation of Visual Absence (VA) neurons and the corresponding probability assigned to the answer "No" for questions in the VA-QA dataset across various LVLMs. We observe that VA neurons consistently exhibit high activation values when the input question contains visually absent tokens, indicating that the models are internally responsive to the absence of relevant visual information. However, the predicted probabilities for "No" remain low, indicating a disconnect between internal detection and output behavior. This discrepancy highlights the necessity for our proposed refinement method using the VA detector, which aims to align the model's response with its internal recognition of visual absence.

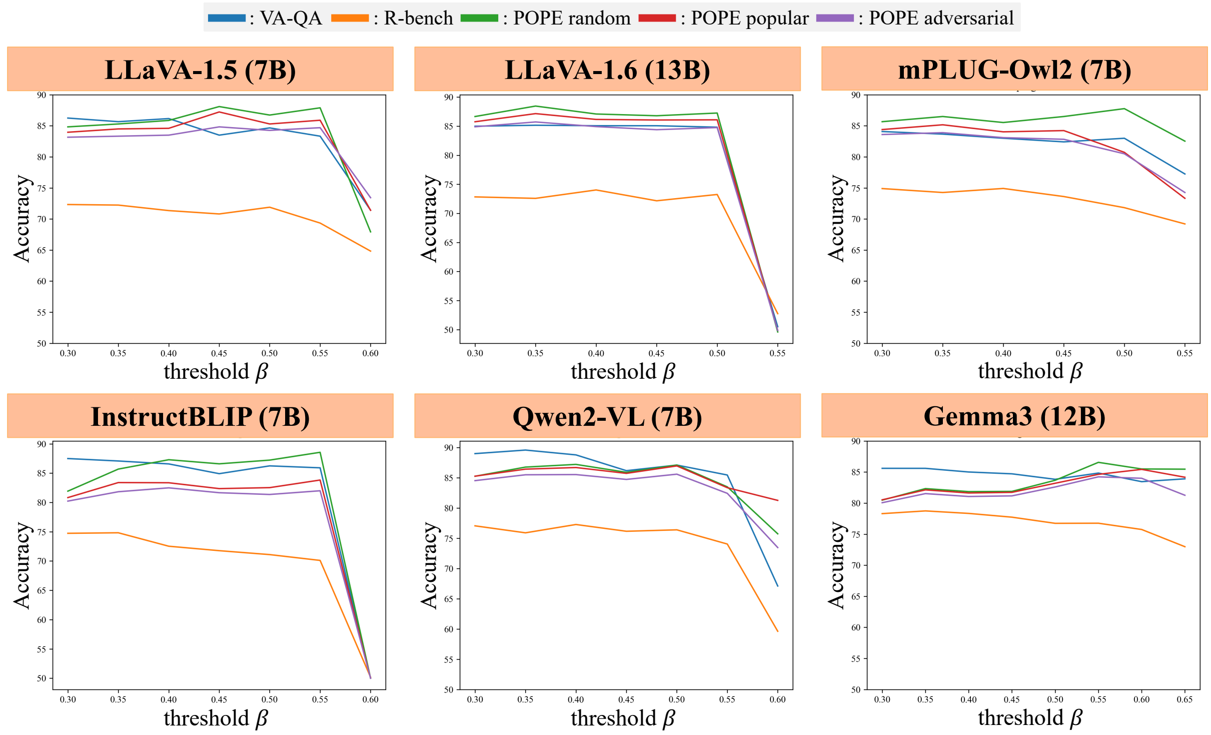


Figure 11: **Effect of Score Threshold β Selection.** This figure shows the accuracy of binary question answering benchmarks across various models depending on the threshold β used to select VA neurons.

C Effect of Score Threshold Selection for VA Neuron Identification

To examine the impact of the threshold hyperparameter β used to identify Visual Absence-aware (VA) neurons, we conduct an ablation study across multiple LVLMs. As described in Section. 4.1, β controls which neurons are selected based on their sensitivity score $\mathbf{S}^{\mathbf{VA}}$, thereby determining the feature set used to train the VA detector.

Figure 11 illustrates the performance of the VA detector across different values of β, evaluated on various binary question answering datasets. We observe a consistent pattern across all models: accuracy peaks within a specific range of β, indicating that the neurons selected at an optimal threshold are most effectively reflecting the model's inherent capability to recognize visual absence. When β is too low, the inclusion of less informative or noisy neurons dilutes the signal. Conversely, overly high thresholds result in too few neurons, weakening the feature representation. The detector trained using the optimal threshold demonstrates stable and high accuracy across all models, demonstrating that a well-chosen subset of VA neurons captures the model's capacity for visual grounding most effectively.

Table 5: **Results on SEED-Bench with VA Detector Trained with Diverse Data.**

Model		**Instance Attributes**			**Visual Reasoning**		
		Acc_{yes}	Acc_{no}	Acc	Acc_{yes}	Acc_{no}	Acc
LLaVA-v1.5 (7B)	Baseline	97.3	36.5	51.7	94.0	46.3	58.2
	Ours	87.5	53.8	62.3	67.2	68.2	67.9
	Ours w/ diverse data	87.2	60.1	**66.9**	73.1	71.1	**71.6**
mPLUG-Owl2 (7B)	Baseline	89.4	58.4	66.2	92.6	60.2	68.3
	Ours	67.8	73.6	72.1	62.7	85.1	79.5
	Ours w/ diverse data	56.9	79.7	**74.0**	65.7	86.1	**81.0**
Qwen2-VL (7B)	Baseline	90.4	71.8	76.5	80.6	69.2	72.0
	Ours	84.1	78.0	79.6	77.6	76.6	76.9
	Ours w/ diverse data	74.4	83.6	**81.3**	73.1	78.6	**77.2**

Table 6: **Results on Binary Question Answering with VA Detector Trained with Diverse Data.**

	In-domain			**Out-of-domain**											
	VA-QA			**R-Bench**			**POPE**								
							Random			Popular			Adversarial		
	Acc_{yes}	Acc_{no}	Acc	Acc_{yes}	Acc_{no}	Acc	Acc_{yes}	Acc_{no}	Acc	Acc_{yes}	Acc_{no}	Acc	Acc_{yes}	Acc_{no}	Acc
Qwen2-VL (7B)	87.3	72.7	80.0	87.9	62.5	75.3	84.6	97.5	90.9	84.6	92.9	88.8	84.6	89.1	86.8
+ Ours	87.6	86.7	87.1	87.0	65.8	76.5	76.5	98.3	87.1	76.5	97.4	87.0	76.5	94.5	85.5
+ Ours w/ diverse data	92.2	80.8	86.5	92.7	58.6	75.8	81.3	97.2	89.0	81.3	94.9	88.1	81.3	91.9	86.6

D Additional Results

D.1 Results on Additional Benchmark Dataset

Table 7: **Results on Winoground.**

Model		Acc_{yes}	Acc_{no}	Acc
LLaVA-v1.5 (7B)	Baseline	90.39	14.00	54.88
	Ours	76.92	48.08	**62.50**
mPLUG-Owl2 (7B)	Baseline	92.31	26.92	59.62
	Ours	55.77	78.85	**67.31**
InstructBLIP (7B)	Baseline	73.08	48.08	60.58
	Ours	44.23	78.85	**61.54**
Qwen2-VL (7B)	Baseline	76.92	51.92	64.42
	Ours	75.00	55.77	**65.38**

We evaluate our method on the Winoground dataset (Thrush et al., 2022), which contains paired images and captions. Similar to our constructed VA-QA dataset (Section. 3.1), each pair of captions in Winoground consists of identical words arranged in different orders. To apply our approach, we convert each caption into a binary question, resulting in two "yes" questions and two "no" questions per pair. We focus on the *Both* split of Winoground, as it contains captions with visually absent words, providing a challenging setting to assess our method. As shown in Table 7, our approach achieves consistent improvements under these conditions.

D.2 Training VA detector with Diverse Data

As we acknowledged in the Limitations section, our VA detector primarily targets object- and relation-level grounding, and we anticipated potential weaknesses when applied to a wider variety of question types. To examine this aspect empirically, we conducted evaluations using two specific categories from SEED-Bench (Li et al., 2023a): "Instance Attributes" for attribute-based hallucinations and "Visual Reasoning" for assessing visual reasoning capabilities. Since SEED-Bench questions are multiple-choice-based, and our method is applicable to binary (yes-or-no) questions, we transformed each multiple-choice question into a series of binary questions as explained in Section. 5.

Interestingly, as illustrated in Table 5, our method effectively handles both attribute-based hallucinations and reasoning-based questions, thereby demonstrating that its generalizability extends beyond the scope we initially expected. Moreover, as mentioned in our Limitations section, we assumed that further training the VA detector on data covering diverse hallucination types would lead to even greater effectiveness. To support this claim experimentally, we train our VA detector on data from each SEED-Bench category mentioned above (with an 8:2 train-test split). The improved results on the test split confirm the benefit of incorporating a wider variety of hallucination types into the training set.

Additionally, a VA detector trained on diverse data also enhances performance on binary benchmark datasets shown in Table 2. Since Qwen2-VL exhibited the most unstable performance, weevaluatede its performance. As shown in Table 6, enriching the training data improved the performance of the VA detector compared to our previous results. Given the clear improvement from adding even a single category of questions, we anticipate that training with a broader and more diverse dataset would yield a more robust and generalizable VA detector across various scenarios.

D.3 Generation with Various Decoding Strategies

Table 8: **Results on Open-ended Generation with Beam Search Decoding.**

Model		$C_s \downarrow$	$C_i \downarrow$	Length	GPT-score $\downarrow$
LLaVA-v1.5 (7B)	Baseline	63.8	30.0	104.5	118.1
	Ours	**62.8**	**29.5**	105.4	**91.8**
mPLUG-Owl2 (7B)	Baseline	67.2	31.8	105.8	135.2
	Ours	**60.8**	**29.1**	110.8	**124.7**
InstructBLIP (7B)	Baseline	57.4	28.7	99.3	124.6
	Ours	**36.6**	**22.4**	116.6	**104.0**
Qwen2-VL (7B)	Baseline	52.0	23.1	253.2	75.6
	Ours	**51.4**	**23.0**	254.9	**73.6**

Our refinement method is independent of decoding strategies as it solely adjusts logit probabilities. It is thus compatible with diverse decoding strategies beyond greedy decoding, including sampling strategies (e.g., setting temperature, top-p, top-k) as well as beam search.

Specifically, in beam search decoding, our method can be directly applied by evaluating each candidate beam individually through the VA detector. When visually absent tokens are detected, the process rolls back to previous tokens within each candidate beam and applies a scoring penalty accordingly. Table 8 demonstrates consistent performance improvements obtained by applying our method in combination with beam search. These results confirm that our approach is robust and decoding strategy-agnostic.

Table 9: **Results on Open-ended Generation with Existing Hallucination-Mitigation Strategies.**

Model		$C_s \downarrow$	$C_i \downarrow$	Length	GPT-score $\downarrow$
LLaVA-v1.5 (7B)	Greedy	60.0	29.0	100.3	111.6
	Greedy + Ours	**58.6**	**28.1**	100.2	**111.3**
	VCD	58.2	28.9	100.6	113.8
	VCD + Ours	**57.6**	**28.6**	100.4	**110.7**
	DoLA	59.0	28.9	99.0	116.9
	DoLA + Ours	**57.0**	**28.4**	98.4	**115.9**
mPLUG-Owl2 (7B)	Greedy	66.8	30.6	105.1	130.7
	Greedy + Ours	**57.2**	**27.9**	104.2	**116.6**
	VCD	68.0	31.7	105.6	134.3
	VCD + Ours	**50.4**	**27.2**	106.9	**111.3**
	DoLA	67.4	30.8	104.6	136.1
	DoLA + Ours	**48.1**	**29.9**	91.6	**82.0**
Qwen2-VL (7B)	Greedy	49.2	**22.6**	252.5	84.4
	Greedy + Ours	**47.2**	23.7	228.2	**83.8**
	VCD	54.2	**24.1**	257.4	84.3
	VCD + Ours	**50.4**	24.4	233.6	**79.7**
	DoLA	43.2	**22.0**	230.0	79.1
	DoLA + Ours	**42.6**	23.1	209.5	**73.1**

D.4 Combination with Existing Hallucination-Mitigation Strategies

Our method specifically aims at maximizing the utility of internally pre-trained knowledge within LVLMs, without relying on external sources or post-training techniques. Our experimental comparisons thus focused primarily on demonstrating improvements in performance relative to their raw outputs, quantifying how much better we can utilize the model's internal knowledge through our approach. Because our method identifies visually absent tokens based on FFN activations of generated tokens during the decoding phase, it can be combined seamlessly with various other hallucination-mitigation strategies, as long as the underlying model terminology remains consistent. To illustrate this point, we present results combining our approach with two representative decoding-based methods, VCD (Leng et al., 2024) and DoLA (Chuang et al., 2023).

For example, when integrated with VCD, we maintain VCD's logit probability calculation procedure; however, when a generated token re-enters as input and is detected as visually absent by our VA detector, the decoding process rolls back one iteration and adjusts the logit probabilities accordingly, prompting the model to generate an alternative token. A detailed description of our open-ended generation refinement method is provided in Section. 4.2 and Appendix F.1.

Table 9 demonstrates consistent improvements achieved when integrating our method with VCD and DoLA, confirming its compatibility and effectiveness with existing training-free hallucination mitigation strategies.

D.5 Evaluation on High-Capacity LVLMs

Table 10 shows clear performance improvements on Qwen2.5-VL (Bai et al., 2025) for both model sizes (7B and 32B), reaffirming the effectiveness of our method on powerful, high-capacity LVLMs and demonstrating the continued relevance of ad-

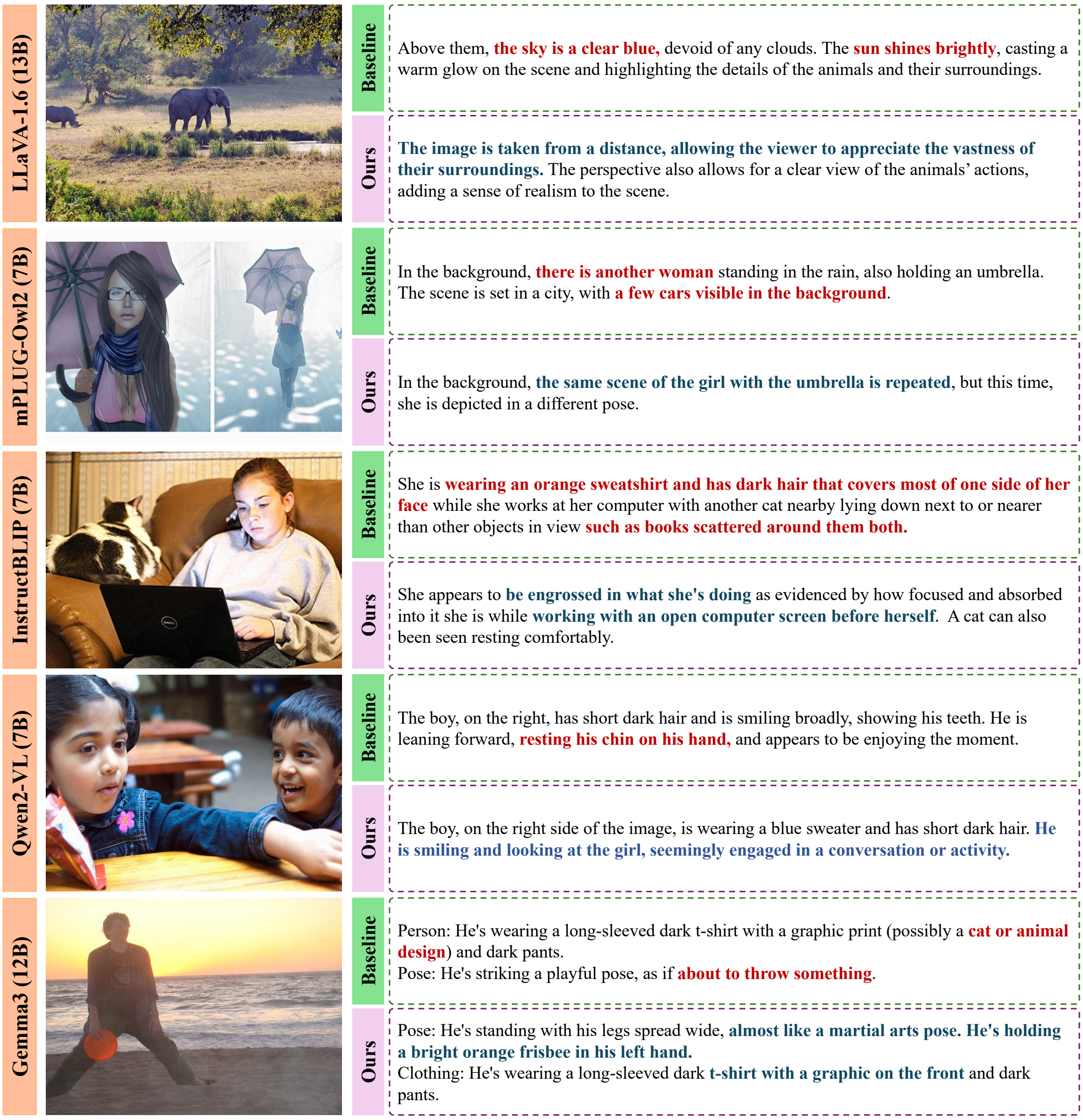
Table 10: **Results on Open-ended Generation of Qwen2.5-VL**

Model		C$_s$ ↓	C$_i$ ↓	Length	GPT-score ↓
Qwen2.5-VL (7B)	Baseline	44.8	23.7	176.1	60.0
	Ours	**35.0**	**23.2**	136.0	**59.8**
Qwen2.5-VL (32B)	Baseline	56.8	24.9	298.9	78.8
	Ours	**47.2**	**24.1**	244.3	**62.0**

dressing vulnerabilities to visually absent tokens.

E Open-ended Generation Qualitative Results

We show the qualitative results of our generation refinement method across various models in Figure 12, excluding the LLaVA-1.5 already presented in the main text. The red colored texts are visually ungrounded words, and the blue colored texts are the words that were revised by our method. As shown in these figures, our method effectively detects visually absent tokens and refines them to words that align with the given image.



LLaVA-1.6 (13B)

Baseline

Above them, **the sky is a clear blue,** devoid of any clouds. The **sun shines brightly**, casting a warm glow on the scene and highlighting the details of the animals and their surroundings.

Ours

The image is taken from a distance, allowing the viewer to appreciate the vastness of their surroundings. The perspective also allows for a clear view of the animals' actions, adding a sense of realism to the scene.

mPLUG-Owl2 (7B)

Baseline

In the background, **there is another woman** standing in the rain, also holding an umbrella. The scene is set in a city, with **a few cars visible in the background**.

Ours

In the background, **the same scene of the girl with the umbrella is repeated**, but this time, she is depicted in a different pose.

InstructBLIP (7B)

Baseline

She is **wearing an orange sweatshirt and has dark hair that covers most of one side of her face** while she works at her computer with another cat nearby lying down next to or nearer than other objects in view **such as books scattered around them both.**

Ours

She appears to **be engrossed in what she's doing** as evidenced by how focused and absorbed into it she is while **working with an open computer screen before herself**. A cat can also been seen resting comfortably.

Qwen2-VL (7B)

Baseline

The boy, on the right, has short dark hair and is smiling broadly, showing his teeth. He is leaning forward, **resting his chin on his hand,** and appears to be enjoying the moment.

Ours

The boy, on the right side of the image, is wearing a blue sweater and has short dark hair. **He is smiling and looking at the girl, seemingly engaged in a conversation or activity.**

Gemma3 (12B)

Baseline

Person: He's wearing a long-sleeved dark t-shirt with a graphic print (possibly a **cat or animal design**) and dark pants.
Pose: He's striking a playful pose, as if **about to throw something**.

Ours

Pose: He's standing with his legs spread wide, **almost like a martial arts pose. He's holding a bright orange frisbee in his left hand.**
Clothing: He's wearing a long-sleeved dark **t-shirt with a graphic on the front** and dark pants.

Figure 12: **Open-ended Generation Qualitative Results.**

F Experimental Details

F.1 Details of VA Detector-based Refinement for Open-ended Generation

We observe that VA neuron activations tend to weaken as the length of generated output increases, which can degrade the model's ability to detect visual absence in open-ended generation tasks. To address this, we mask previously generated sentences except the most recent one when extracting activations. Additionally, if the model repeatedly generates a visually absent token at the same position, we gradually roll back towards earlier tokens until it produces a more accurate and visually grounded response. Specifically, we progressively deepen the rollback by an additional step each time two rollbacks occur within a 5-token window. All experiments presented in the paper employ greedy text decoding to ensure determinism and eliminate randomness.

F.2 Implementation Details and Fit Quality of VA Detector

To construct the Visual Absence-aware Detector, we utilized sklearn (Pedregosa et al., 2011) packages to train an MLPClassifier with a single hidden layer of 128 units. This detector was trained on features derived from VA neurons. From the original VA-QA dataset of 2400 samples, we first curated a subset consisting exclusively of instances answered correctly by each model individually. This curated subset was then partitioned into training and validation sets using a 9:1 ratio. To identify the optimal threshold hyperparameter β, which is used to select VA neurons, we search values from 0.3 to 0.8 in increments of 0.05, selecting β that maximizes the accuracy on the validation set. For all evaluations on downstream tasks, we employed greedy decoding to ensure deterministic outputs.

To quantitatively assess the performance of our VA detector, Table 11 illustrates the fit quality of our VA detector for each LVLM, including metrics such as precision, recall, and accuracy. Despite being trained on a small amount of data with a lightweight 2-layer MLP classifier, the results demonstrate strong fitting quality across all metrics. This confirms that the activation patterns of VA neurons exhibit consistency and robustness across different models.

Table 11: **Fit Quality of VA detector.**

Model	Precision	Recall	Accuracy
LLaVA-v1.5 (7B)	0.965	0.943	97.6
LLaVA-v1.6 (13B)	0.940	0.979	97.6
mPLUG-Owl2 (7B)	0.938	0.958	97.0
InstructBLIP (7B)	0.980	0.978	98.8
Qwen2-VL (7B)	0.951	0.934	97.2
Gemma3 (12B)	0.993	0.929	97.4

Table 12: **Accuracy on Role-based Questions.**

Model	Acc
LLaVA-v1.5 (7B)	90.0
mPLUG-Owl2 (7B)	98.0
Qwen2-VL (7B)	96.0

F.3 Robustness of VA Detector

When constructing the dataset used to train our VA detector, we explicitly excluded ambiguous cases, selecting only clearly and definitively visually absent concepts (Figure 2). Thus, our VA detector is specifically tuned to detect only tokens that are unambiguously absent from the visual content, avoiding detection of ambiguous or partially grounded tokens.

Furthermore, there might be cases where the questions contain visually absent tokens yet require a "Yes" answer, so we constructed an additional evaluation dataset specifically designed for these cases. This dataset covers role-based questions: these questions inquire about the role or purpose of objects in the image. For instance, if an image shows a hanger without any clothes and the question asks, "Is this object used for hanging clothes?"—though "clothes" is visually absent, the correct answer remains "Yes."

As demonstrated in Table 12, our method achieves strong performance in these scenarios as well, highlighting its robustness to such nuanced cases.

F.4 Details of Evaluation Datasets

VA-QA (our constructed dataset) Based on SVO-probe, we construct a set of yes-or-no questions for each image using the corresponding ⟨subject, verb, object⟩ triplet. As detailed in Section. 3.1 : (1) Positive questions (answer=Yes) directly utilize the provided triplet, accurately describing the image. (2) Negative questions (answer=No) are formed by altering one element of the original triplet to create a mismatch with the image, thereby differing only in that single modification from the

“Yes” question.

R-bench (Wu et al., 2024) Unlike POPE, which only targets object hallucination, R-bench is a dataset for assessing hallucinations related to inter-object relationships. Based on MSCOCO (Lin et al., 2014) captions, they extract relationship triplets by applying a scene graph parser and convert them into a yes-or-no question. The negative question (answer = No) is constructed by varying one element from the triplet. We only use the balanced subset, which has an equal number of positive and negative questions, for unbiased evaluation.

POPE (Li et al., 2023b) The Polling-based Object Probing Evaluation (POPE) dataset is designed to assess ’object’ hallucination in LVLMs. It evaluates the model’s ability to answer yes-or-no questions in the form: “Is there a ⟨object⟩in the image?”. For objects detected in the image, questions are constructed with the answer “Yes”. Conversely, questions with the answer “No” are divided into 3 variants-*Random*, *Popular*, and *Adversarial*- based on the frequency and co-occurrence of the absent objects.

CHAIR (Rohrbach et al., 2018) The Caption Hallucination Assessment with Image Relevance (CHAIR) metric is designed to quantify object hallucination in generated image captions. It operates by comparing object words mentioned in a caption against the ground truth object annotations of the corresponding image. Specifically, CHAIR measures the proportion of hallucinated object words, which are defined as object words in the caption that do not appear in the image’s ground truth annotation list. A lower CHAIR score indicates less hallucination and thus better caption quality in terms of object grounding.

F.5 GPT-4o assisted Open-ended Generation Evaluation

Even though CHAIR is the most widely used metric for evaluating object existence, it has limitations: it does not cover all objects in the image and fails to address other types of hallucination, such as those related to attributes or relations. Therefore, we employ GPT-4o (Hurst et al., 2024), a leading multi-modal model, to assist in our generation evaluation. As shown in Table 13, we design a prompt that mimics the CHAIR evaluation process. This prompt guides the model to identify both visually grounded and hallucinative elements within the generated captions, while also scaling the severity of each identified hallucination. For each data, we extract a list of severity scores for the hallucinative elements, and calculate an individual hallucination score s_{hall} for each data by summing the generated list of severity scores. Then we determine the final reported metric by taking the mean of these values across all data and multiplying it by 10.

```
<ASSISTANT> {caption generated by the model}

<INSTRUCTION> We would like to request your feedback on the performance of the AI assistantino image caption generation of the given image.

Please evaluate the <ASSISTANT>'s caption against the image by listing the elements in two bullet-point categories:

Correct Elements: a list of phrases in the caption that are directly and objectively grounded in the image.

Hallucinated Elements: a list of phrases that describe visual content that does not appear in the image. For each hallucinated phrase, add a severity score from 1 to 5 in parentheses.
  Use the following scale:
    - 1 (Minimal): Slight over-description or plausible extrapolation; not directly visible but weakly implied.
    - 2 (Low): Mild misstatement; plausible but not quite accurate.
    - 3 (Moderate): Clearly ungrounded; some viewers may misinterpret presence based on image context.
    - 4 (High): Definitely not in the image, and no contextual cues suggest it.
    - 5 (Severe): Fabricated object or scene element with strong visual contradiction — impossible or clearly absent.

Output correct and hallucinated elements in a **flat list format**([]), separately (do not output in bullets). Do not add additional explanation.
```

Table 13: **Prompt template for GPT-4o for generation evaluation.** We use the GPT-4o's ability to evaluate the quality of the generation, by instructing it to generate correct elements and hallucinated elements in the caption generated by the model. We further use this result to make a hallucination score.