VisiPruner: Decoding Discontinuous Cross-Modal Dynamics for Efficient Multimodal LLMs

Yingqi Fan[1], Anhao Zhao[1,3], Jinlan Fu[5], Junlong Tong[1,2], Hui Su[4], Yijie Pan[1], Wei Zhang[1], Xiaoyu Shen[1]*

[1]Ningbo Key Laboratory of Spatial Intelligence and Digital Derivative, Institute of Digital Twin, EIT, Ningbo [2]Shanghai Jiao Tong University
[3]Hong Kong Polytechnic University [4]Meituan Inc. [5]National University of Singapore

Correspondence: yingqi949@gmail.com xyshen@eitech.edu.cn

Abstract

Multimodal Large Language Models (MLLMs) have achieved strong performance across vision-language tasks, but suffer from significant computational overhead due to the quadratic growth of attention computations with the number of multimodal tokens. Though efforts have been made to prune tokens in MLLMs, *they lack a fundamental understanding of how MLLMs process and fuse multimodal information.* Through systematic analysis, we uncover a **three-stage** cross-modal interaction process: (1) Shallow layers recognize task intent, with visual tokens acting as passive attention sinks; (2) Cross-modal fusion occurs abruptly in middle layers, driven by a few critical visual tokens; (3) Deep layers discard vision tokens, focusing solely on linguistic refinement. Based on these findings, we propose *VisiPruner*, a training-free pruning framework that reduces up to 99% of vision-related attention computations and 53.9% of FLOPs on LLaVA-v1.5 7B. It significantly outperforms existing token pruning methods and generalizes across diverse MLLMs. Beyond pruning, our insights further provide actionable guidelines for training efficient MLLMs by aligning model architecture with its intrinsic layer-wise processing dynamics. Our code is available at: https://github.com/EIT-NLP/VisiPruner.

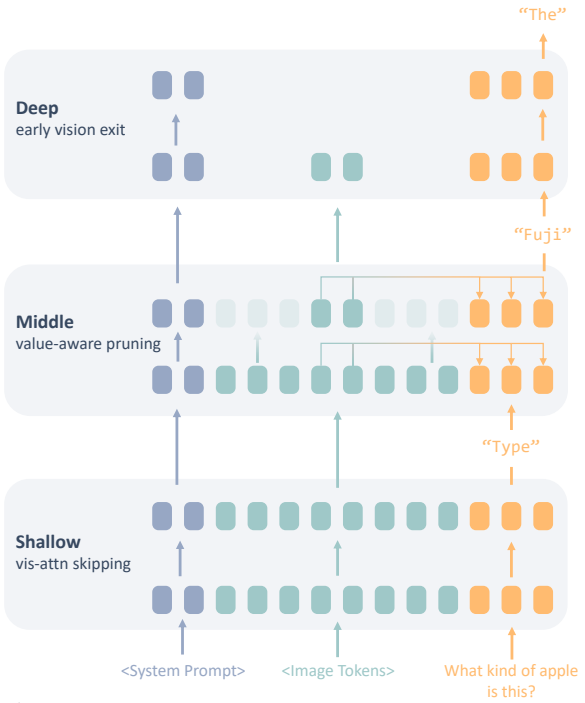


Figure 1: **Illustration of the three-stage discontinuous information processing in Multimodal Large Language Models (MLLMs).** The framework separates visual-text integration into three key stages: **Shallow Layers** focus on task recognition, **Middle Layers** highlight the cross-modal fusion of sparse, task-relevant visual tokens, and **Deep Layers** focus on linguistic refinement after vision integration.

1 Introduction

Multimodal Large Language Models (MLLMs) (Yin et al., 2024) extend the reasoning power of Large Language Models (LLMs) to other modalities like vision (Li et al., 2023a), audio (Guzhov et al., 2021), and video (Alayrac et al., 2022; Tong et al., 2025b), typically by aligning modality encoders (e.g., ViT (Dosovitskiy et al., 2021)) with LLMs through lightweight projectors (Liu et al., 2023a; Lin et al., 2025; Chen et al., 2025a; Zhao et al., 2024). However, visual encoders often produce far more tokens than text due to higher information density. This not only inflates the sequence length but also results in a quadratic increase in attention computation. While recent efforts like token pruning (Ye et al., 2024a; Shang et al., 2024; Lin et al., 2024), dynamic resolution (Arif et al., 2024; Li et al., 2024), and sparse attention mechanisms (Zhang et al., 2024b; Li et al., 2025) aim to mitigate this issue, their effectiveness remains limited due to *a fundamental gap in understanding how MLLMs actually process and integrate visual information across layers.*

Existing analyses of cross-modal interactions in

*Corresponding Author

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 18897–18914
November 4-9, 2025

MLLMs predominantly rely on attention scores as proxies for information flow (Wu et al., 2024; Zhang et al., 2025a, 2024c). This has led to widespread but misleading conclusions, e.g., the assumption that cross-modal fusion mainly occur in shallow layers. We move beyond attention maps to understand how and when visual information is actually utilized, revealing three insights that revise the current understanding of MLLMs:

- **Shallow Layers as Task Recognizers**: Contrary to prior beliefs (Wu et al., 2024; Zhang et al., 2025a, 2024c), cross-attention in early layers serves *no meaningful role* in visual-text fusion. Visual and textual tokens evolve *independently*, with shallow layers functioning solely to recognize task from text instructions, while visual tokens act merely as 'attention sinks" (Xiao et al., 2024).

- **Sparse Critical Tokens in Middle Layers**: Cross-modal integration occurs abruptly in intermediate layers, but only *a few critical visual tokens drive this process*. Conventional attention-based methods fail to identify these tokens, as their importance correlates with feature similarity rather than attention weights.

- **Instruction Alignment in Deep Layers**: Once visual information has been integrated into the text encoder, deeper layers *discard vision tokens* and transition to pure linguistic refinement to output final answers.

Building on these insights, we introduce *VisiPruner*, a training-free pruning framework that exploits both layer-wise and token-wise redundancy in MLLMs. For layer-wise compression, our method disables cross- and self-attention in shallow visual layers and removes visual tokens in deep layers, allowing seamless integration with existing token pruning methods. For token-wise compression, we propose a novel influence-based method to dynamically identify and retain only the most interactive visual tokens from middle layers. Together, these strategies reduce up to 99.0% of visual-related attention computations and 53.9% of total FLOPs, all while preserving performance across a range of MLLMs and benchmarks.

Our findings further offer actionable guidelines for designing efficient MLLMs. While *VisiPruner* demonstrates the principles in a training-free paradigm, embedding them directly into MLLM training pipelines should further optimize performance-efficiency tradeoffs. Overall, our work makes four key contributions: (1) To the best of our knowledge, we are the first systematic analysis revealing the discontinuous, sparse, and decoupled nature of cross-modal interactions in MLLMs, particularly highlighting the counter-intuitive finding that *shallow layers operate independently of vision*; (2) Exposing the inadequacy of attention-based analysis for understanding visual token utility by attention merging; (3) A training-free pruning framework validated across diverse MLLMs and benchmarks; and (4) Actionable guidelines for designing efficient MLLMs that align with their intrinsic mechanics.

2 Background

Modern MLLMs integrate perceptual modalities (e.g., vision) with linguistic reasoning using a vision encoder, projection, and language backbone (Liu et al., 2023b; Chu et al., 2024).

Modality-Specific Encoding Let input $v \in \mathcal{V}$ (e.g., an image) and text instruction $x \in \mathcal{X}$. Each modality is encoded independently:

$$\text{Visual encoder:} \quad \mathbf{E}_v = \mathcal{V}(v) \in \mathbb{R}^{N_v \times d_v},$$
$$\text{Textual encoder:} \quad \mathbf{E}_t = \mathcal{T}(t) \in \mathbb{R}^{N_x \times d_x},$$

where $\mathcal{V}$ (e.g., ViT) and $\mathcal{T}$ (i.e., LLM tokenizer) map inputs to sequences of embeddings. Typically, $N_v \gg N_x$ due to the high information density of $\mathcal{V}$, e.g., $N_v = 576$ for a 336×336 image with patch size 14 (Chen et al., 2024a).

Cross-Modal Projection A projector $\mathcal{P}$ aligns visual embeddings to the LLM's text space $\mathbf{H}_x^{(0)}$:

$$\mathbf{H}_v^{(0)} = \mathcal{P}(\mathbf{E}_v) \in \mathbb{R}^{N_v \times d_h},$$

where d_h matches the LLM's hidden dimension.

Layer-Wise Cross-Modal Fusion The fused input is defined as $\mathbf{H}^{(0)} = \mathbf{H}_v^{(0)} \oplus \mathbf{H}_t^{(0)}$ ($\oplus$ denotes concatenation), which is processed through L transformer layers. At layer l, cross-attention and self-attention are computed as (Zhao et al., 2025):

$$\mathbf{Q}_t^{(l)} = \mathbf{H}_t^{(l-1)}\mathbf{W}_Q, \quad \mathbf{K}_v^{(l)}, \mathbf{V}_v^{(l)} = \mathbf{H}_v^{(l-1)}\mathbf{W}_{K/V},$$

$$\mathbf{A}^{(l)} = \text{softmax}\left(\frac{\mathbf{Q}_t^{(l)}(\mathbf{K}_v^{(l)})^\top}{\sqrt{d_h}}\right) \in \mathbb{R}^{N_x \times N_v},$$

$$\mathbf{H}_{\text{cross}}^{(l)} = \mathbf{A}^{(l)}\mathbf{V}_v^{(l)},$$

$$\mathbf{H}_t^{(l)} = \text{TransformerBlock}(\mathbf{H}_t^{(l-1)} + \mathbf{H}_{\text{cross}}^{(l)}),$$

A key computational bottleneck in MLLMs arises from the large number of visual tokens (Zhang et al., 2025a). In most scenarios, $N_v \gg N_x$, the cross-attention matrix $\mathbf{A}^{(l)} \in \mathbb{R}^{N_x \times N_v}$ grows significantly, making its computation a dominant cost factor. We believe that not all visual tokens contribute meaningfully to text-driven reasoning. To address this, we seek to (1) *Understand cross-modal information flow* and (2) *Reduce unnecessary visual-text interaction*.

3 Shallow Layers: Task Recognition

Shallow layers in MLLMs are often assumed to be crucial for cross-modal fusion due to two observations: (1) High cross-attention scores between instruction tokens and vision tokens in early layers (Wu et al., 2024; Zhang et al., 2025a); and (2) Performance degradation when cross-attention in shallow layers is masked (Zhang et al., 2024c). We systematically re-evaluate these claims and present evidence that contradicts these assumptions.

3.1 Attention Scores ≠ Information Utility

Although attention scores are often interpreted as measures of token importance, we provide two key counterarguments that challenge this assumption.

Counterpoint 1: Static Attention Patterns We first visualize attention maps across shallow, middle and deep layers (App. D). A striking pattern emerges: the most attended vision tokens remain unchanged regardless of the input instruction in shallow layers. Whether the task involves color identification (e.g., "What color is the dog?") or scene understanding (e.g., "Is there any scooter?"), the same image regions consistently receive the highest attention. Although it is counterintuitive that different tasks attend to the same visual features, these visual tokens may contribute to global understanding of the image (Darcet et al., 2024).

Counterpoint 2: Masking Highly Attended Tokens has No Effects To further test if highly attended tokens encode global information, we mask the top 10% most attended vision tokens in layers 1–2 and evaluate performance. If these tokens were essential, their removal should degrade performance. However, results show minimal change (Tab. 7). This directly contradicts the claim that attention scores reflect information utility. Apparently, *high attention scores in shallow layers do not imply high information utility*.

Model	GQA	MME^P	POPE	MMB
LLaVA-v1.5 7B	62.0	1507.6	85.9	64.3
+ Mask	62.0	1506.6	85.7	64.3
LLaVA-v1.5 13B	63.3	1531.3	85.9	67.7
+ Mask	63.2	1518.6	86.3	68.9
InternVL2.5 8B	63.6	1700.0	90.6	84.6
+ Mask	63.2	1689.5	90.6	84.3
MobileVLM-v2 3B	61.0	1440.5	84.7	63.2
+ Mask	60.9	1440.8	84.6	63.3

Table 1: Performance after masking top 10% attended visual tokens in the first two layers on diverse MLLMs. See App. B for results under different selection criteria.

3.2 Redundant but Necessary?

Given that high attention ≠ information utility, we now examine whether shallow-layer visual tokens serve any information utility at all.

The Redundancy Paradox Following our previous result that masking top 10% most attended tokens has no effect, if cross-modal fusion does occur in shallow layers, it would have to reside in the remaining 90% of tokens. We now mask the remaining 90% tokens to see if these tokens alone are sufficient for multimodal fusion. Surprisingly, we again find minor degradation in overall accuracy[1] (72.6 → 71.5, suggesting that *neither the most attended nor the least attended vision tokens carry essential information!* To further probe the necessity of individual tokens, we randomly mask half of the visual tokens and measure performance changes:

- **Left Half Masking** (removing first 288 of 576 tokens): 72.6 → 72.6
- **Right Half Masking** (removing last 288 tokens): 72.6 → 72.4.

We can see that the performance remains stable regardless of which tokens are masked, implying that *visual tokens in shallow layers are largely redundant in terms of content transfer*.

During decoding, we observe the same as in pre-filling stage (see Sec. 3.4), confirming the absence of cross-modal fusion in shallow layers. It

[1]Averaged over four benchmarks (GQA, MME, POPE and MMB) and two MLLMs (LLaVA-v1.5 7B and 13B). Note that the score for MME is divided by 20 before averaging.

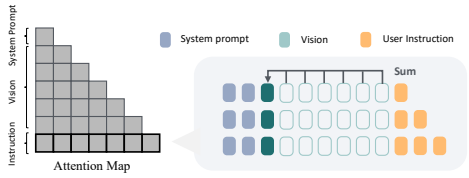


Figure 2: Merge visual attention weights into a single token to stabilize the attention distribution of the first layer.

is likely that visual tokens in shallow layers do not contribute to information fusion in a meaningful way. Instead, their presence—regardless of which specific tokens remain—appears to be necessary for stability rather than content transfer.

3.3 Vision as Attention Stabilizers

Given that no individual vision tokens carry essential cross-modal information, it is paradoxical that cross attention knockout in shallow layers leads to significant loss in visual perception (Zhang et al., 2024c; Geva et al., 2023). Hence, we hypothesize that their role is to stabilize shallow-layer attention distributions without transmitting meaningful content. To test this, we propose *Attention Merging*, forcing all cross-attention weights in shallow layers to focus on a single visual token (Fig. 2):

$$\mathbf{A}_{i,j}^{(l)} = \begin{cases} \sum_{v \in V} A_{i,v}^{(l)} & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where V represents all vision tokens and k is the randomly selected index of the merged token. If shallow vision tokens were performing useful fusion, constraining attention to a single token should degrade performance. However, we observe no meaningful change across different choices of k (see App. G), confirming that *no specific vision token is necessary for shallow-layer computation*. The model simply requires *some* tokens to absorb attention weights. Even further, we show that the stabilization is needed only in the first layer:

- **layer 1**: masking all vision tokens significantly degrades average performance ($72.6 \rightarrow 65.2$), confirming that a visual attention sink is needed.
- **layer 2-7**: system prompts can replace vision tokens as attention sinks, with no performance drop ($72.6 \rightarrow 72.1$) (see App. E).

This dichotomy arises from diverging value vector distributions: early vision token values ($\mathbf{V}_v^{(0)}$) differ significantly from text tokens ($\mathbf{V}_x^{(0)}$), necessitating modality-specific sinks initially (App. F).

Overall, these results suggest that *shallow-layer vision tokens primarily serve as a stabilization mechanism for attention, rather than contributing to meaningful cross-modal fusion of information*.

3.4 Null Effects in Decoding Stage

The cross-modal fusion happens in two stages:

- **Prefill Phase**: The entire input sequence, including visual and text embeddings, is processed in a single forward pass. This initializes hidden states for subsequent decoding.
- **Decoding Phase**: Tokens are generated autoregressively, where each new token attends to previously generated tokens while interacting with visual representations.

Apart from the prefilling stage, we also remove vision tokens from the key-value (KV) cache at different depths in the decoding stage. As can be seen in Tab. 2, the result is even better after removing the KV cache (see App. J). This further supports our claim that shallow visual tokens do not meaningfully contribute to content information. Instead, their role appears to be largely structural rather than informational.

Model	Layers	MM-Vet	GQA
LLaVA-v1.5 7B	-	31.2	62.0
	1–8	33.8	61.8
	9–15	28.3	61.8
	26–32	31.1	61.9
	1–32	26.1	61.7

Table 2: **Performance with visual information removed from specific KV Cache layers.** MM-Vet is a benchmark requiring key visual information to remain in the KV Cache (Yu et al., 2024).

3.5 Role of Shallow Layers

Having confirmed the absence of meaningful cross-modal information flow, and the visual and text layers evolve largely independently. We further investigate the actual roles of shallow layers.

Shallow Text Layers: Task Recognition To analyze what shallow text layers are mainly doing, we analyze the semantic content of the final token's hidden state by projecting it through the model's unembedding matrix (nostalgebraist, 2020):

$$D_{\text{last}} = \text{softmax}(W_u h_{\text{last}}^{\ell}), \quad (2)$$

where D_{last} represents the probability distribution over vocabulary tokens. We find that shallow-layer representations align with task semantics

rather than visual content. For example, intermediate layers produce activations aligned with task-relevant words: - "*How many cars...*" → "number" (Layer 10) - "*What kind of...*" → "type" (Layer 7) (App. H).

Beyond the latent representation of the final input token, we further observe that the value-output matrix also encodes task information in shallow layers, reinforcing our finding (App. I).

$$D_{\text{vo}} = \text{softmax}(W_u \cdot V^{\ell}_{\text{last}} \cdot O), \tag{3}$$

These findings suggest that *shallow text layers are primarily responsible for task recognition, operating independently from visual processing*.

Shallow Visual Layers: Feature Alignment Knowing that little cross-modal interaction is performed in shallow visual layers, we further investigate whether intra-modal fusion occurs. Specifically, we mask self-attention among visual tokens, forcing each token to be processed independently. As shown in Tab. 3, this modification results in only a minimal performance drop, indicating that self-attention plays a negligible role.

These results suggest that *the primary function of shallow visual layers is neither cross, nor intra-modal fusion, but rather the alignment of ViT features with the LLM's internal representation space, implying that the attention mechanism in these layers may be largely redundant*.

Layers	Masking	#Token	Merging	GQA
-	-	N/A	N/A	61.95
1-2	C	576	No	57.41
		575	Yes	**61.98**
	C&V	576	No	56.08
		575	Yes	**61.96**
1-7	C	576	No	57.18
		575	Yes	**61.51**
	C&V	576	No	54.63
		575	Yes	**60.78**

Table 3: **Impact of vision on cross-attention stability.** *Layers* refer to layers with attention masked. *# Tokens* indicates the number of masked vision tokens. "C" represents cross attention masking; "V" represents visual self attention masking.

3.6 Strategy for Efficient MLLM Design

Based on these findings, we propose a simple yet effective pruning strategy for shallow layers: (1) Merge visual attention in layer 1 to serve as an attention sink; (2) Skip visual-textual attention computation for all vision tokens in layers 2+; and (3) Remove visual self-attention.

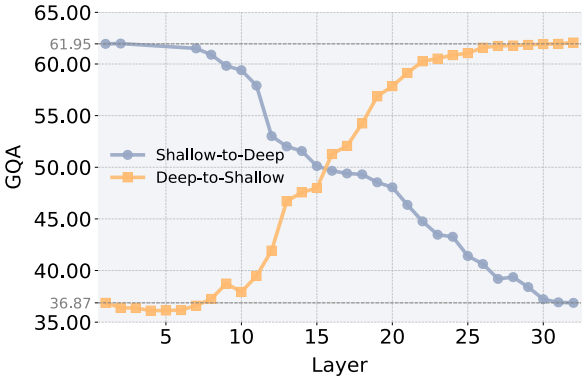


Figure 3: Masking ranges of layers, from shallow-to-deep and deep-to-shallow, exhibit a clear reduction in cross-modal fusion at both shallow and deep layers.

4 Middle Layers: Sparse Grounding

Beyond certain stage, we find that fully masking cross-attention begins to significantly deteriorate performance again from around 9th layer as shown in Fig. 3, suggesting a transition into middle layers.

4.1 Confirming Cross-Modal Fusion

Given our prior analysis of shallow layers, this performance drop may also result from disruptions in the attention distribution rather than cross-modal interaction, so we perform two key analyses: *attention merging and key visual token masking.*

Re-examine Attention Merging We examine the impact of attention merging (Sec. 3.3) in middle layers . Compared to simple cross-attention masking, attention merging results in worse performance with GQA on LLaVA-v1.5 7B: 61.95 → 51.73 → 49.42, suggesting that the drop is not merely due to attention distribution disruption.

Key Visual Token Masking Next, we examine whether middle-layer attention is instruction relevant. We mask the top and bottom 10% attended visual tokens for comparison in layers 9–15:

- **Top 10% tokens**: GQA 61.95 → 54.09
- **Bottom 10% tokens**: GQA 61.95 → 61.93

The significant performance drop when masking highly attended visual tokens, compared to the negligible impact of masking least-attended tokens, suggests that in the middle layers, cross-attention is focused on instruction-relevant regions, *confirming meaningful cross-modal fusion* in these layers.

4.2 Sparsity of Cross-Modal Fusion

Given that middle layers are fusing visual features, we explore this fusion requires all visual tokens or

only a sparse subset of them.

Selective Vision Masking We apply cross-attention-based selection, retaining only the top 5% most attended tokens unmasking, discarding remaining 95%. The model still maintains a comparable performance (72.6 $\rightarrow$ 71.3), confirming that middle layers start to focus on a sparse subset of vision tokens, rather than the entire image.

Visual Focus Tracking While each middle layer may shift its focus to different visual regions when searching for the answer, we visualize the locations of critical visual tokens on the image and find that the model consistently focuses on instruction-relevant regions across layers (see App. K).

These results imply that (1) *cross-modal fusion in middle layers is sparse, only a few critical visual tokens are required*; and (2) *critical visual tokens stay unchanged across layers, there is no need to re-identify critical tokens at each layer*.

4.3 Identifying Critical Visual Tokens

Regarding this sparsity, we aim to develop a method that accurately identifies these critical visual tokens to reduce complexity. The most straightforward method is based on cross-attention weights. However, we find this approach is limited by (1) *Visual Attention Sink Tokens*: The visual attention sink phenomenon is present across all layers, introducing irrelevant tokens in attention-based selection; (2) *Difficulty Isolating Single Token Influence*: Attention weights are distributed across all tokens, which can introduce uncertainty when isolating the impact of individual tokens; and (3) *Static Thresholds on Tokens Number*: Attention-based selection requires setting a fixed threshold, which reduces flexibility across different tasks.

Another intuitive approach to measure the influence of each vision token is to mask them individually and observe their effect on the final output. However, this requires propagating changes through all layers, making it computationally expensive. Instead, we propose a more efficient method that directly evaluates the impact of each vision token on the attention output of the last input token, which determines the first answer token.

Attention Computation Recap The attention weight matrix is calculated as:

$$W = \text{softmax}(\frac{QK^T}{\sqrt{d_k}} + M) \tag{4}$$

where Q, K are the query and key matrices, W is the attention weight,and M is a causal mask. The attention output is then computed by:

$$O = \text{Reshape}(\sum_{heads} W \cdot V) \tag{5}$$

where V is the value matrix, O the attention output.

Token Masking Procedure To evaluate the influence of token j on token i at layer ℓ, we modify the attention weight matrix as follows:

$$W'_{i \to j} = 0 \tag{6}$$

which masks the ability of token i to attend to token j across all attention heads. Using this masked attention weight, we recompute the attention output:

$$O' = \text{Reshape}(\sum_{heads} W'_{i \to j} \cdot V) \tag{7}$$

Influence Measurement The influence of token j on token i is quantified by comparing the original attention output of token i and the masked attention output of token i using two complementary metrics: cosine similarity and L2 distance.

We measure the directional similarity between the original and masked outputs:

$$\text{Cosine Similarity}_{i \leftarrow j} = \frac{O_i \cdot O'_{i\ \text{masked}}}{\|O_i\|_2\ \|O'_{i\ \text{masked}}\|_2}.$$

where $\|\cdot\|_2$ is the L2-norm. A lower similarity indicates a stronger influence of token j on token i, as masking token j significantly alters the output.

In addition to directional changes, we also measure the magnitude of change using the L2 distance:

$$\text{L2 Distance}_{i \leftarrow j} = |O_i - O'_{i,\text{masked}}|_2. \tag{8}$$

A larger L2 distance reflects a greater impact of token j on token i, as it quantifies the absolute difference in output magnitude after masking.

By combining cosine similarity and L2 distance, we capture both directional and magnitude-based influences of vision tokens, offering a better way to identify **the most critical tokens** than using attention weights (See Tab. 4 for detailed comparison).

4.4 Strategy for Efficient MLLM Design

Given the sparsity of cross-modal fusion in middle layers, we propose an adaptive, training-free pruning strategy that retains only the most influential vision tokens: If masking a vision token reduces the cosine similarity below 0.995, we define this layer as a filtering layer, implying the visual input

starts to contribute to the answer generation. Then, at this filtering layer, we discard vision tokens with a L2 distance below 0.2, as they have a negligible impact on the last input token. Using this method, we prune 576 vision tokens down to an average of 10.3 after the filtering layer, maintaining competitive performance with only a 0.7% drop in GQA. Moreover, our middle-layer pruning offers a new interpretability lens on vision token redundancy by lowering the minimum visual tokens retained.

Strategy	POPE	GQA	VQA^{T}	MMVet
Attn[2](last)	85.9	60.3	57.1	25.4
Attn (text)	85.9	58.0	55.6	23.8
Attn (vis)	85.9	55.2	52.0	20.9
Value-aware	**86.1**	**61.3**	**57.8**	**31.9**

Table 4: Value-aware pruning in middle layers consistently outperformances attention-based methods, particularly in multi-token generation tasks like GQA, TextVQA and MMVet, indicating a stronger ability to retain instruction-relevant visual information.

5 Deep Layers: Linguistic Alignment

As seen in Figure 3, we observe that beyond certain layers, masking all cross-attention connections once again has minimal impact on performance, which indicates a transition to deep layers.

5.1 Discontinuous Vision Dependence

To explore the role of vision tokens in different layers, we compare the performance impact of discarding visual tokens versus skipping visual processing only at specific layers. This allows us to better understand when vision tokens can be discarded.

Skipping $\neq$ Discarding When we discard all visual tokens from layer 20 and beyond, we observe a noticeable drop in performance on the GQA dataset, from 61.95 to 59.13. However, when we only skip the visual processing at layer 20 and allow visual information to continue through subsequent layers, the performance degradation is minimal, from 61.95 to 61.66. This suggests that while visual tokens remain relevant beyond layer 20, the processing in this layer itself is not essential. Therefore, we conclude that *vision dependence may not be continuous*. Specifically, skipping one layer of visual processing does not necessarily imply that skipping all subsequent layers yields the same.

[2]Top 10 visual tokens most attended by the final text token, instruction tokens and visual tokens at layer 16.

Layers	Top words in vocabulary space
32	The, In, All, """, There, **L**, **Lux**, I A, It, **Lux**
31	The, All, In, """, There, **L**, the, all A, It, **Lux**
30	All, The, all, **Lux**, the, In, **lux**, **L** A, It, There
25	**Lux**, **lux**, all, scene, the, scene, Scene The, A, It, There

Figure 4: Top vocabulary tokens from the semantic projection of the last input token at each layer.

Discarding in Deep Layers Next, we investigate the impact of discarding visual tokens from layer 26. On GQA, we observe negligible performance change, from 61.95 to 61.91, indicating that the visual information processed in earlier layers is already sufficiently integrated. However, when we skip visual processing at layer 26 and allow subsequent layers to process the visual information, the performance drops more significantly, from 61.95 to 61.40. This suggests that by layer 26, visual tokens have already been integrated into the textual representation, and the visual information starts to introduce noise or redundancy in later layers.

Further supporting this, we observe minimal performance loss when masking cross-attention in deeper layers (Fig. 3), as well as when removing vision from the deep-layer KV Cache (Sec. 3.4). These results reinforce the idea that *after a certain layer, vision tokens can be safely discarded without significantly affecting performance*.

5.2 Behavior: Linguistic Alignment

Using the prompt "*What are all the scene text in the image?*", we project the hidden state of the last input token to the semantic space (Eq. 2). As shown in Fig. 4, by layer 25, the model generates the correct visual answer "*Lux*", but struggles to structure it into a coherent response, "*The scene text is '**Luxmi Jewellers**'.*" While visual content is correctly identified, it is initially misplaced linguistically. As we move to deeper layers, the model gradually refines the output, prioritizing tokens "*The*" to form a grammatically correct sentence.

These findings suggest that *deep layers are responsible for aligning the generated content with natural language conventions*.

5.3 Strategy for Efficient MLLM Design

Having known that deep layers no longer rely on vision tokens, we introduce a pruning strategy to detect the completion of vision-to-text fusion: After identifying and the only retaining critical vision tokens from middle layers (Sec. 4.4), we con-

Models	FLOPs(T)	Method	GQA	SQA^I	VQA^T	POPE	MME^P	MMB	MMStar	Avg. ↑
LLaVA-v1.5 7B	3.82	dense	62.0	66.8	58.2	85.9	1507.6	64.3	33.7	63.8
	1.76	ours	60.3	66.7	55.2	84.4	1428.3	62.0	33.3	61.9
LLaVA-v1.5 13B	7.44	dense	63.3	71.6	61.3	85.9	1531.3	67.7	36.2	66.1
	3.31	ours	61.3	**72.0**	59.1	84.7	1485.3	66.9	36.0	64.9
InternVL-v2.5 8B	11.00	dense	63.6	98.0	79.1	90.6	1680.8	84.6	60.4	80.1
	5.34	ours	58.8	97.8	77.7	88.2	1643.5	79.6	59.9	77.0
QwenVL-v2 7B	9.62	dense	62.4	85.4	76.9	87.9	1687.7	79.4	56.3	64.6
	4.69	ours	62.2	84.1	74.4	87.8	1615.8	78.0	77.9	63.5
MobileVLM-v2 3B	0.37	dense	61.0	70.0	57.5	84.7	1440.5	63.2	35.1	63.5
	0.25	ours	57.6	69.4	53.5	81.7	1402.3	57.4	**36.7**	63.3

Table 5: **Performance of *VisiPruner* across various MLLMs and benchmarks.** These benchmarks include visual question answering datasets GQA (Hudson and Manning, 2019), MME (Fu et al., 2024), MMBench (Liu et al., 2024), and MMStar (Chen et al., 2024b), visual reasoning benchmark SQA (Lu et al., 2022), OCR benchmark TextVQA (Singh et al., 2019), and the object hallucination benchmark POPE (Li et al., 2023b).

Method	Vis Attn Computation	MMB	SQA^I	GQA	MME^P	VQA^T	POPE	MMVet	Avg.
LLaVA-v1.5 7B	100.0%	64.3	66.8	62.0	1507.6	58.2	85.9	31.2	63.4
PDrop $_{\text{retained=192}}$	−86.4%	63.2	70.2	57.1	1419.8	56.1	82.3	30.5	61.5
SparseVLM $_{\text{retained=192}}$	−86.4%	64.1	68.7	59.5	1441.1	56.1	85.3	33.1	62.7
FastV$_{\text{k=3,r=0.75}}$	−87.3%	63.5	68.7	57.5	1458.9	56.2	81.0	27.9	61.1
PDrop $_{\text{retained=64}}$	−97.6%	33.3	69.2	41.9	982.3	45.9	55.9	30.7	46.6
SparseVLM $_{\text{retained=64}}$	−97.6%	60.1	69.8	53.8	1351.4	53.4	77.5	24.9	58.2
FitPrune $_{\text{reduction=0.9}}$	−98.0%	55.4	67.8	52.4	1210.2	52.1	60.5	24.2	53.3
Ours	−98.3%	62.0	66.7	60.3	1428.3	55.2	84.4	29.1	61.3

Table 6: **Compare *VisiPruner* with training-free token-wise compression baselines**, including: **FastV** (Chen et al., 2024a), which keeps tokens selected by the last-to-vision attention; **FitPrune** (Ye et al., 2024b), which prunes tokens according to attention-distribution saliency; **SparseVLM** (Zhang et al., 2025b), which drops tokens based on cross-attention importance; and **PyramidDrop** (Xing et al., 2024), which progressively reduces visual tokens.

tinuously track their influence. If these kept tokens show no measurable impact for two consecutive layers, we define the latter layer as the **vision exit layer** (ℓ_{exit}). Beyond ℓ_{exit}, those retained vision tokens are removed, further eliminating redundant computations. On LLaVA-v1.5 7B, this method identifies an average vision exit at layer 23.9, while still maintaining the performance on GQA 62.0 → 61.3 → 61.0, confirming that deep layers operate independently of vision.

6 $\mathcal{V}isi\mathcal{P}runer$ and Future MLLMs

Based on key insights into the role of vision tokens and cross-modal interactions within LLaVA-v1.5 7B, this section aims to (1) validate the generalization ability of our conclusions across diverse MLLMs and (2) provide actionable recommendations for future model design.

Generalization Ability We apply our analytical methods and pruning strategies to multiple MLLMs with different architectures, including LLaVA-v1.5 13B, MobileVLM-V2-3B (Chu et al., 2024), Qwen2-VL 7B (Wang et al., 2024) and InternVL2.5-8B (Chen et al., 2025b). InternVL2.5 and Qwen2-VL are recently released MLLMs that dynamically generates image tokens, allowing us to verify the scalability of our conclusions in models with more flexible visual processing. MobileVLM 3B is a compact model with significantly fewer image tokens, enabling us to test the applicability in a MLLM with less parameters.

Complexity Analysis By eliminating visual-relevant attention in shallow layers and deep layers while adaptively pruning to 10 vision tokens in middle filtering layers, we reduce cross-modal attention operations to minimal levels, achieving 98.3% reduction in visual-related attention computation and a 53.9% reduction in FLOPs compared to baseline. Building on our vision-independent layer identification, we maintain only the most interactive vision tokens on average in middle layers while completely excluding visual tokens from KV

caching in shallow and deep layers. This strategic retention reduces the original visual KV cache memory and further lowers computational overheads in long-sequence decoding scenarios. Details about FLOPs calculation are in App. M.

Method Comparison Given that our method disables visual attention in shallow layers, we use the visual attention FLOPs reduction ratio as the evaluation criterion to ensure a fair comparison. Notably, our layer-wise compression strategy is compatible with token pruning approaches and can further reduce computational overhead through shallow-layer visual attention merging and early vision exit.

Suggestions for Future MLLMs Based on our findings, we propose several guidelines to improve the efficiency and interpretability of future MLLMs: *(a) Truncate shallow visual layers and eliminate cross/self-attention* Since shallow layers contribute little to cross-modal fusion, computational overhead can be reduced by processing visual tokens only up to the middle layers. The model can be trained to recognize the start of middle layers, or adapted to a fixed starting point. *(b) Train models to attend sparsely* By training for sparse attention in middle layers, the model directly identifies critical tokens, bypassing the need for post-hoc attention scores or influence measurements. *(c) Enable early exiting in deep visual layers once modality fusion is established.* Given the established linguistic alignment behavior in deep layers, we recommend incorporating vision exit mechanisms into MLLM training pipelines to automatically skip out when fusion is finished.

7 Conclusions

We propose a three-stage MLLM framework—where shallow layers handle intra-modal task interpretation, middle layers integrate task-relevant visual tokens into textual embeddings, and deep layers focus on linguistic alignment. Building on these insights, we introduce stage-specific optimizations that boost computational efficiency, and validated our framework across multiple MLLM architectures, confirming its general applicability. Finally, we distill our findings into practical guidelines for future MLLM design.

Limitations

While our study provides a principled and general framework for understanding the mechanisms of vision-language models, there are several limitations. First, training the projector to align vision tokens with semantic representations and inserting them until later layers could further strengthen our findings regarding intra-modal processing in shallow layers. Second, due to hardware constraints, our analysis was limited to models with up to 13 billion parameters. Future work could replicate our approach using larger models, potentially uncovering additional insights through our three-stage analytical framework.

Acknowledgement

We thank EIT and IDT High Performance Computing Center for providing computational resources for this project. This work was supported by the 2035 Key Research and Development Program of Ningbo City under Grant No.2024Z123 and No. 2025Z034.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. *Preprint*, arXiv:2204.14198.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. 2024. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. *Preprint*, arXiv:2408.10945.

Haoran Chen, Junyan Lin, Xinhao Chen, Yue Fan, Xin Jin, Hui Su, Jianfeng Dong, Jinlan Fu, and Xiaoyu Shen. 2025a. Rethinking visual layer selection in multimodal llms. *Preprint*, arXiv:2504.21447.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024a. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. *Preprint*, arXiv:2403.06764.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. 2024b. Are we on the right way for evaluating large vision-language models? *Preprint*, arXiv:2403.20330.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, and 23 others. 2025b. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *Preprint*, arXiv:2412.05271.

Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, and Chunhua Shen. 2024. Mobilevlm v2: Faster and stronger baseline for vision language model. *Preprint*, arXiv:2402.03766.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2023. Analyzing transformers in embedding space. *Preprint*, arXiv:2209.02535.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. 2024. Vision transformers need registers. *Preprint*, arXiv:2309.16588.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *Preprint*, arXiv:2010.11929.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2024. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Preprint*, arXiv:2306.13394.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. *Preprint*, arXiv:2304.14767.

Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. 2021. Audioclip: Extending clip to image, text and audio. *Preprint*, arXiv:2106.13043.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Preprint*, arXiv:1902.09506.

Hongliang Li, Jiaxin Zhang, Wenhui Liao, Dezhi Peng, Kai Ding, and Lianwen Jin. 2025. Beyond token compression: A training-free reduction framework for efficient visual processing in mllms. *arXiv preprint arXiv:2501.19036*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Preprint*, arXiv:2301.12597.

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

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2024. Monkey: Image resolution and text label are important things for large multi-modal models. *Preprint*, arXiv:2311.06607.

Junyan Lin, Haoran Chen, Yue Fan, Yingqi Fan, Xin Jin, Hui Su, Jinlan Fu, and Xiaoyu Shen. 2025. Multi-layer visual feature fusion in multimodal llms: Methods, analysis, and best practices. *Preprint*, arXiv:2503.06063.

Junyan Lin, Haoran Chen, Dawei Zhu, and Xiaoyu Shen. 2024. To preserve or to compress: An in-depth study of connector selection in multimodal large language models. *Preprint*, arXiv:2410.06765.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *Preprint*, arXiv:2304.08485.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2024. Mmbench: Is your multi-modal model an all-around player? *Preprint*, arXiv:2307.06281.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. 2023c. Scissorhands: Exploiting the persistence of importance hypothesis for llm kv cache compression at test time. *Preprint*, arXiv:2305.17118.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Preprint*, arXiv:2209.09513.

Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. 2024. Towards interpreting visual information processing in vision-language models. *Preprint*, arXiv:2410.07149.

nostalgebraist. 2020. Interpreting gpt: The logit lens.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2024. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *Preprint*, arXiv:2403.15388.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. *Preprint*, arXiv:1904.08920.

Junlong Tong, Jinlan Fu, Zixuan Lin, Yingqi Fan, Anhao Zhao, Hui Su, and Xiaoyu Shen. 2025a. Llm as effective streaming processor: Bridging streaming-batch mismatches with group position encoding. *Preprint*, arXiv:2505.16983.

Junlong Tong, Wei Zhang, Yaohui Jin, and Xiaoyu Shen. 2025b. Context guided transformer entropy modeling for video compression. *arXiv preprint arXiv:2508.01852*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

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. *Preprint*, arXiv:2409.12191.

Qiong Wu, Wenhao Lin, Weihao Ye, Yiyi Zhou, Xiaoshuai Sun, and Rongrong Ji. 2024. Accelerating multimodal large language models via dynamic visual-token exit and the empirical findings. *Preprint*, arXiv:2411.19628.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. *Preprint*, arXiv:2309.17453.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. 2024. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *Preprint*, arXiv:2410.17247.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. 2024a. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. *Preprint*, arXiv:2409.10197.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. 2024b. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. *Preprint*, arXiv:2409.10197.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, 11(12).

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2024. Mm-vet: Evaluating large multimodal models for integrated capabilities. *Preprint*, arXiv:2308.02490.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. 2025a. Llava-mini: Efficient image and video large multimodal models with one vision token. *Preprint*, arXiv:2501.03895.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. 2024a. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *Preprint*, arXiv:2410.04417.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. 2025b. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *Preprint*, arXiv:2410.04417.

Zeliang Zhang, Phu Pham, Wentian Zhao, Kun Wan, Yu-Jhe Li, Jianing Zhou, Daniel Miranda, Ajinkya Kale, and Chenliang Xu. 2024b. Treat visual tokens as text? but your mllm only needs fewer efforts to see. *arXiv preprint arXiv:2410.06169*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H_2o: Heavy-hitter oracle for efficient generative inference of large language models. *Preprint*, arXiv:2306.14048.

Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. 2024c. Cross-modal information flow in multimodal large language models. *Preprint*, arXiv:2411.18620.

Anhao Zhao, Fanghua Ye, Yingqi Fan, Junlong Tong, Zhiwei Fei, Hui Su, and Xiaoyu Shen. 2025. Skipgpt: Dynamic layer pruning reinvented with token awareness and module decoupling. *Preprint*, arXiv:2506.04179.

Anhao Zhao, Fanghua Ye, Jinlan Fu, and Xiaoyu Shen. 2024. Unveiling in-context learning: A coordinate system to understand its working mechanism. *Preprint*, arXiv:2407.17011.

A Related Work

A.1 Cross-modal Information Flow in MLLMs

Research on cross-modal information flow in MLLMs has shown that visual information is gradually integrated into the generation of subsequent textual tokens (Neo et al., 2024; Wu et al., 2024; Zhang et al., 2024c; Tong et al., 2025a). However, there is still disagreement about how and when this fusion occurs within the model. Neo et al. (2024) suggest that key visual information is primarily extracted in the middle to late layers of the model. In contrast, based on attention weight analysis, Wu et al. (2024) and Zhang et al. (2025a) argue that visual information is fused into textual tokens in the shallow layers, highlighting the role of vision tokens early in the process. Similarly, Zhang et al. (2024c) report that the model is constantly fusing visual information, starts with perceiving the entire image and then extracting key visual details.

A.2 In-VLM Vision Compression

Identifying and retaining important tokens that are crucial for generation is a key aspect of effective training-free token pruning (Xiao et al., 2024; Zhang et al., 2023; Liu et al., 2023c). To make vision compression more adaptive to user instructions, in-VLM compression has become a key area of research. Chen et al. (2024a) observe the significant of redundancy of vision tokens via the sparsity of attention for vision tokens within VLMs, and propose a pruning method named FastV to pick the most important vision tokens based on attention each vision token received from the last token. Building on FastV, PyramidDrop drops vision tokens in multiple stages (Xing et al., 2024). SparseVLM selects visual-relevant text tokens to evaluate the importance of vision tokens based on the self-attention matrix, then prunes the vision tokens using a rank-based strategy and token recycling to maximize sparsity while retaining essential information (Zhang et al., 2024a).

B Mask Highly Attended Visual Tokens in Shallow Layers Using Different Selection Criteria

To further validate that highly attended visual tokens has no effects, we conducted experiments on additional selection criteria:

Criterion	GQA	MME^P	VQA^T	POPE
vanilla	62.0	1507.6	58.2	85.9
attn (last → vis)	62.0	1506.6	57.9	85.7
attn (text → vis)	62.0	1503.6	58.1	85.7
pos (near text)	62.0	1501.1	58.1	85.7

Table 7: Performance after masking top 60 attended visual tokens in the first two layers using different selection criteria.

C Comparison of Cross-Attention Masking Across Different Stages

We also compare the performance on different different benchmarks with cross attention masked in different stages as shown in Tab. 8. The shallow and deep layers exhibit significantly cross-modal information fusion compared with middle layers.

Model	Layers	GQA	MME^P	VQA^T
LLaVA-v1.5 7B	Dense	62.0	1507.6	58.2
	1–7	61.5	1411.2	56.8
	9–15	51.7	722.6	51.1
	27–32	61.8	1488.5	58.1

Table 8: Performance on Various Benchmarks with Cross-Attention Masked in Specific Layers.

D Visualization of visual attention sink phenomenon

In Fig. 5, we visualize the attention distribution on the input image across shallow, middle and deep layers to highlight the visual attention sink phenomenon. Ideally, attention distribution should adapt dynamically based on the input, directing focus to different areas for different tasks. However, our visualizations reveal an intriguing pattern: tokens with high attention scores—highlighted in the image—tend to appear consistently in the same regions across various instructions **in both shallow and deep layers**. This finding suggests that certain vision tokens act as attention sinks, drawing focus but failing to provide meaningful contributions to the model's reasoning. As a result, these tokens may not be essential for generating accurate responses.

Moreover, in the middle layers, we observe that the model starts to concentrate its attention on the more instruction-relevant areas. This reinforces our conclusion that MLLMs undergo a three-stage information processing approach, where shallow layers focus on task recognition, middle layers selectively fuse instruction-relevant visual information, and deep layers refine and align the response with the instruction.

Another interesting finding is that the first layer exhibits clear attention window, the lower half of vision tokens receive more attention from the last input token.

E Detailed Analysis on Visual Attention Sink Tokens

E.1 Lower L1 Norm of Value Vectors for Sink Tokens

As shown in the lower subplot of Fig. 7, visual sink tokens with high attention weights exhibit significantly lower magnitudes in their value vectors. This suggests that visual sink tokens function similarly to textual sink tokens, acting as bias terms in the softmax computation.

E.2 Attention Redistribution After Removing Visual Sink Tokens

After identifying the visual sink tokens in an example, we remove these tokens before the first layer. We observe that the attention weight previously allocated to the visual sink tokens is redistributed to the textual sink tokens in the system prompt.

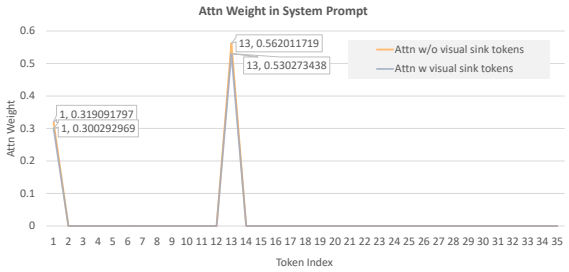
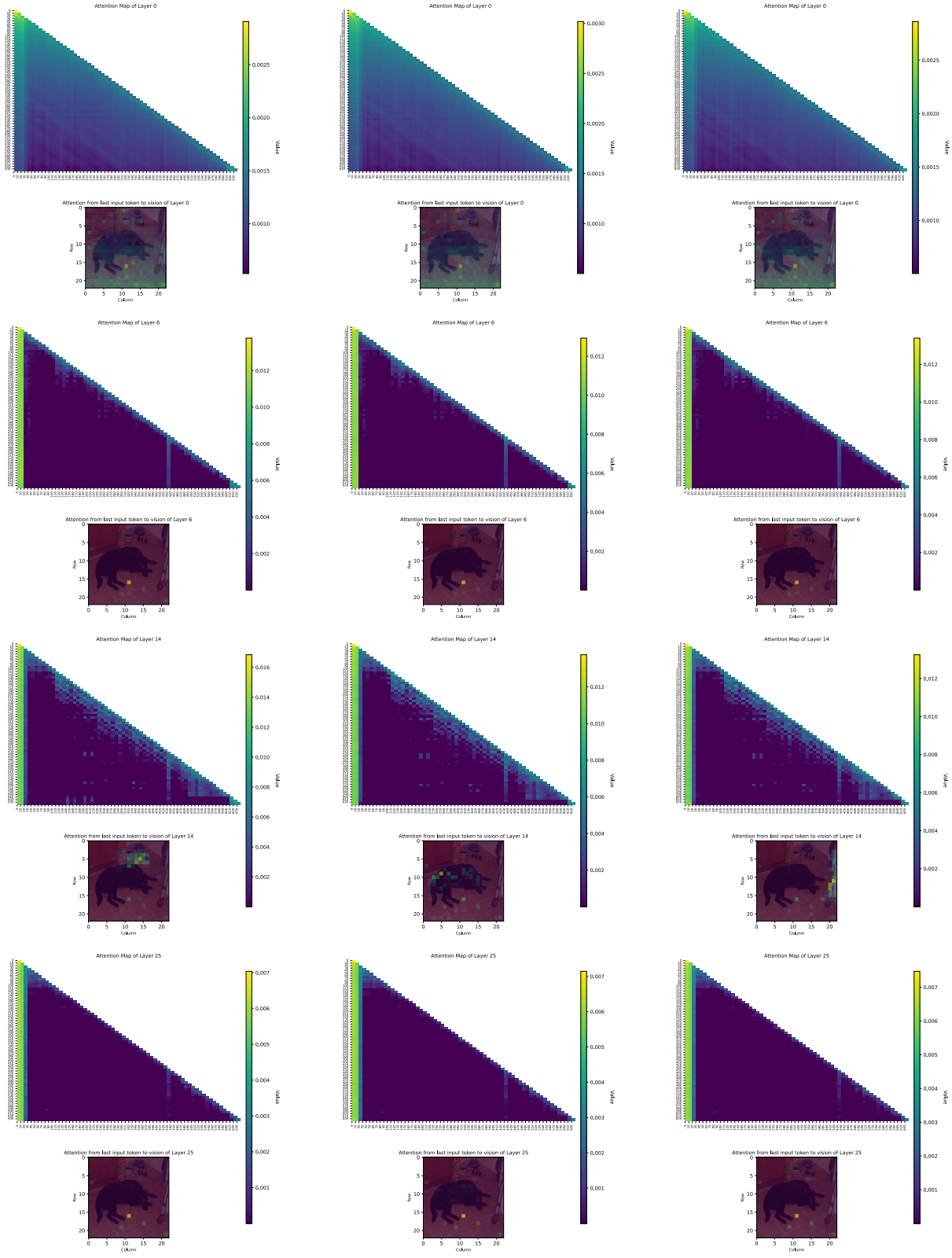


Figure 6: Textual sink tokens in the system prompt absorb the attention weight when visual sink tokens are removed in the third layer.

Sum of attention weight from visual sink tokens: 0.053352. Difference in attention weight of textual sink tokens with and without visual sink tokens: 0.050537109.

F L1 Norm of Value Vectors

As illustrated in Fig. 7, the value vectors for textual and visual tokens show distinct patterns in the first layer. This likely indicates that the model differentiates between modalities at this stage, highlighting the necessity of modality-specific sinks.



(a) USER: How many bowls are there in the image?

(b) USER: What color is the dog in the image?

(c) USER: Is there any scooter in the image?

Figure 5: Visualization of attention map and distribution on image with different instruction across shallow, middle and deep layers using LLaVA-v1.5 7B

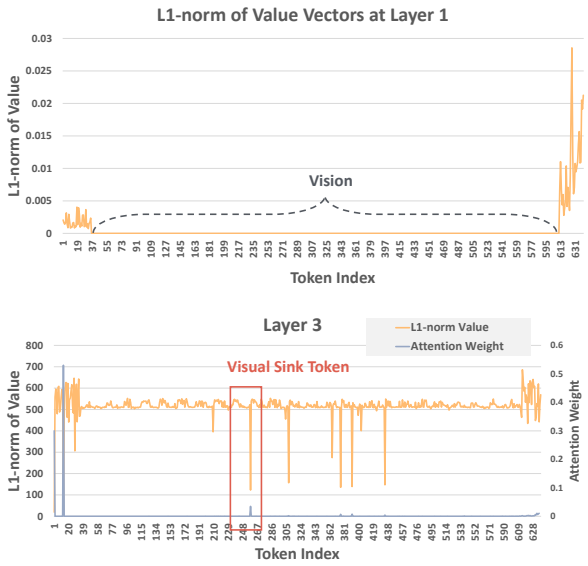


Figure 7: Visualization of attention map and distribution on image with different instruction across shallow, middle and deep layers using LLaVA-v1.5 7B

G Random Selection of Visual Attention Merging Token

To ensure the visual token selection for merging is not index-dependent, we randomly choose a visual token and merge all visual cross-attention into it.

Visual Token Index	GQA
vanilla	61.95
-	57.41
1	61.98
576	61.55
128	61.83
288	61.76

Table 9: Performance of random visual token merging on GQA.

H Complete Results on Semantic Projection of the Last Input Token

In this section, we present a more detailed analysis of the semantic projection of the last input token for different user instructions.

H.1 USER: How Many Cars Are in the Image?

As shown in Tab. 11, when given the user instruction "*How many cars are there in the image?*", the model accurately identifies it as a number-related task.

H.2 USER: What Kind of Apple Is This?

As shown in Tab. 12, when given the user instruction "*What kind of apple is this?*", the model correctly identifies it as a type-related task.

I Task Recognition: Projection of Value-Output Matrix on Semantic Space

The value-output matrix plays a key role in in-context learning by summarizing task-related information. Building on the approach from (Dar et al., 2023), we project this matrix into the semantic space as follows:

$$D = W_u(V_{last} \cdot O) \tag{9}$$

where V is the value vector, O is the output matrix, and W_u is the word unembedding matrix.

I.1 USER: Where is the place of origin?

Given the instruction "*Where is the place of origin?*", the model recognizes this as a location-related task Tab. 13.

Layer	Head	Top words in vocabulary space
14	31	names,Names,NAME,ját,Names
13	31	location,locations,map,Location,Map
12	31	thy,thee,thou,Gemeins,Tu

Table 13: Top 5 tokens from the semantic projection of the value-output matrix of the last input token at different layers.

I.2 USER: How many apples are there in the image?

Given the instruction "*How many apples are there in the image?*", the model recognizes this as a counting-related task Tab. 14.

Layer	Head	Top words in vocabulary space
13	31	two,another,deux,atori,three
12	31	counting,counts,numbers,count,count
11	31	你,your,you,vous,yourself

Table 14: Top 5 tokens from the semantic projection of the value-output matrix of the last input token at different layers.

I.3 USER: What is the make of the car on the left?

Given the instruction "*What is the make of the car on the left?*", the model recognizes this as a brand-related task Tab. 15.

Model	Layers	Vision			Text		Math	Overall
		Recognition	OCR	Spatial awareness	Knowledge	Generation	Math	
LLaVA-v1.5 7B	Dense	36.1	23.9	26.3	17.1	22.4	11.5	31.2
	0–7	39.5	25.2	28.8	21.4	26.9	15.4	**33.8**
	8–14	34.0	21.4	26.5	16.1	19.0	7.7	29.2
	25–31	35.9	22.2	23.2	18.6	22.4	11.2	31.1
	0–31	33.1	13.5	23.5	14.2	16.6	7.7	26.1

Table 10: **Performance Breakdown of LLaVA-v1.5 7B on MM-Vet with Vision Removal from Specific Layers in the KV Cache.** "*Layers*" column indicates the layers from which visual information was removed.

Layers	Top words in vocabulary space
19	four, three, five, several, six, many, seven two, Several, dozen
18	four, three, several, two, five, dozen, lots, many **number**, multiple
17	four, three, several, two, dozen, five, **number** mehrere, lots, multiple
16	four, three, **number** two, five, dozen, several many, mehrere, lots
15	four, **number**, three, Ges, dozen, several, lots five, **count**, multiple
14	four, **number**, three, Ges, two, érique, **count** lots, There, ieri
13	**number**, three, **count**, **number**, four, érique none, ocker, multip, estaven
12	**number**, arden, rita, **Number**, multip, three **NUM**, licz, **number**, **NUM**
11	**number**, arden, rita, **Number**, none, licz **number**, Sa, three, Ges
10	**number**, arden, rita, ubre, nim, konn, eben multip, 兴, two
9	**number**, rita, multip, nim, arden, platz, iken zero, un, VS

Table 11: Top tokens from the projection of the last input token at each layer.

Layers	Top words in vocabulary space
9	sterd, publique, **typen**, Hinweis, penas, ohl, bpe Hero, Sob, ermeister
8	sterd, **typen**, publique, paździer, 庄, schrift 泉, intrag, penas, Hinweis
7	sterd, penas, quelle, **typen**, 泉, teil, wohl paździer, 庄, intrag
6	sterd, paździer, strij, sierp, kwiet, penas, ści Wikispecies, wohl, konn

Table 12: Top tokens from the projection of the last input token at each layer.

Layer	Head	Top words in vocabulary space
14	31	different,Wat,isse,iesen,newer
13	31	brand,companies,company,Brand,brand
12	31	loro,ihnen,your,their,nx

Table 15: Top 5 tokens from the semantic projection of the value-output matrix of the last input token at different layers.

J Analysis of Vision Removal Impact on MM-Vet Performance in KV Cache

To further probe the role of shallow layers, we conducted a vision removal experiment using MM-Vet, a benchmark requiring extended responses where key visual information must be preserved in the KV Cache. Specifically, we examined whether the model relies on vision information from shallow layers during the decoding process. A detailed breakdown of MM-Vet with vision removal on specific layers to determine whether performance degradation or improvement is attributed to vision or text generation. After pruning visual information from the first eight layers, the model performed better than the original configuration, further consolidating that the model does not utilize visual information from shallow layers (see Tab. 10). Additionally, removing vision tokens in deep layers also have a minimal influence on the performance, indicating that the model focuses on processing textual information to align with instruction.

K Visualization of Instruction-Relevant Focus Across Middle Layers

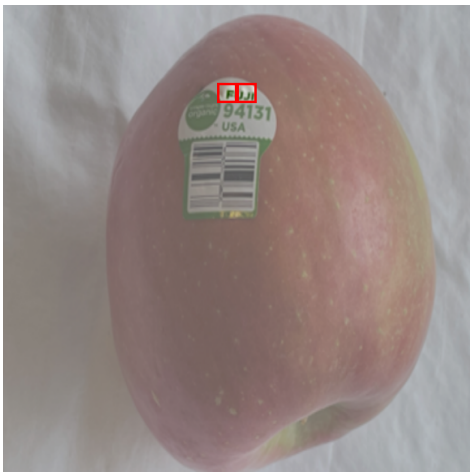


Figure 8: **The Most Instruction-Relevant Region Highlighted in Red Boxes.**

Given the user instruction "*What kind of apple is this?*" and the image in Fig. 8, we observe that the last token in the middle layers consistently focuses on the most instruction-relevant region (see Tab. 16).

Layers	Top 10 Visual Tokens Indices
22	**107**, **108**, 129, 130, 60, **222**, 155, 255, 512, 162
21	**107**, **108**, 129, 130, 60, **222**, 155, 255, 512, 162
20	**107**, **108**, 60, 162, 161, **222**, 163, 61, 399, 255
19	**108**, **107**, 60, **222**, 255, 387, 399, 61, 207, 299
18	**108**, **222**, **107**, 207, 60, 502, 155, 88, **355**, 399
17	**107**, **222**, **108**, 155, 60, 512, 130, 156, 255, 129
16	**107**, **108**, **222**, 155, 60, 156, 131, **355**, 109, 340
15	**107**, **108**, **222**, 60, 61, 255, 88, 163, 399, 155
14	**222**, **107**, **355**, **108**, 340, 159, 574, 255, 398, 131
13	**222**, **107**, **355**, **108**, 340, 398, 574, 255, 60, 155
12	**222**, **355**, 340, 398, 270, 155, 574, **107**, 272, 207
11	**222**, **355**, 340, 574, 575, 398, **108**, **107**, 155, 156
10	**222**, 575, **355**, 574, 340, 398, 207, 571, 272, **108**

Table 16: Top 10 most attended vision tokens from the last input token at each layer. Green indicates the most critical visual tokens, while red marks the visual attention sink tokens.

L Layer-wise Cross-Attention Masking on MobileVLM 3B

Compared to LLaVA-v1.5 7B, MobileVLM v2 3B has a broader range of shallow layers and fewer deep layers. This suggests that smaller models may require more computations on task recognition.

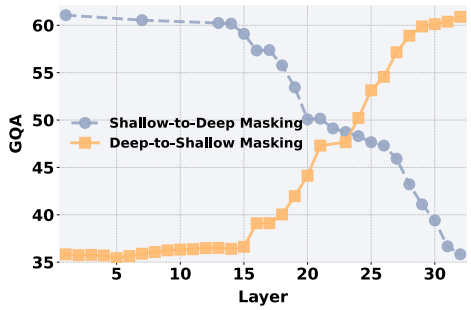


Figure 9: Impact of masking layer ranges from shallow-to-deep and deep-to-shallow, showing a clear reduction in cross-modal fusion in both shallow and deep layers.

M FLOPs Analysis on LLaVA-1.5 7B

Our proposed method greatly reduces vision-related self-attention, cross-attention and FFN, leading to an overall FLOPs reduction of $> 60\%$. Here is a detailed analysis:

The total computation in MLLMs primarily consists of two components: attention computation and feed-forward network (FFN) computation. Among these, attention computation scales quadratically with sequence length, making it the primary computational bottleneck—especially in models like Qwen2-VL, which can generate up to 12,000 visual tokens. For instance, in LLaVA-1.5 7B, the FLOPs for attention computation can be expressed as $2n^2d$. The reduction ratio for visual attention computation is given by:

$$R = 1 - \frac{L'2 * 2(n'_v)^2 d + L'(n'_v n_t)d}{32 * (2(n_v^2 d + n_v n_t))}$$

where the L' the number of cross-modal interaction layers, n'_v represents the number of retained visual tokens. If the input sequence consists of 650 tokens (576 visual tokens and 74 text tokens), our approach eliminates attention computation in shallow and deep layers, retaining only a few critical tokens for cross-modal fusion. This results in a 99% reduction at maximum in attention computation.

FLOPs Calculation. In LLaMA 2 7B (Touvron et al., 2023), the primary flops include FFN and self-attention. The flops for FFN is $3ndm$, where n is the number of input tokens, d is the hidden state size, and m is the intermediate size of the FFN. Hence, the FLOPs overall calculation for visual tokens follows:

$$\sum_{i=0}^{L_{\text{middle}}} \left(4n'_v d^2 + 2n'^2_v d + 3n'_v dm\right) + \sum_{i=0}^{L_{\text{shallow}}} \left(4n_v d^2 + 3n_v dm\right)$$

This optimization leads to an overall visual FLOPs reduction of 62.8% under the given setting (576 visual tokens and 74 text tokens), significantly enhancing efficiency while maintaining performance. Given that the efficiency gain scales with longer textual or visual inputs, our pruning framework offers much greater benefits for longer text instructions or when multiple images are provided.

Additionally, following our actionable guidelines for optimizing MLLMs, the visual computation overhead within shallow layers in FFN should be able to be further reduced through training.

N Failure Case Analysis

In this section, we present an analysis on failure cases in GQA, where our pruned model produced

1,125 mismatched answers compared to the vanilla LLaVA-v1.5 7B over 12,000 samples.

- 234 answers were correct in our model but incorrect in the vanilla model.
- 325 answers were incorrect in our model but correct in the vanilla model.

Upon closer inspection, we found that misclassifications were often related to **variations in word choice** rather than fundamental misunderstandings. Below are some examples:

N.1 "Which kind of vehicle is in front of the flag?\nAnswer the question using a single word or phrase."

- **Ground Truth Answer**: "van"
- **Vanilla Model**: "truck"
- **Ours**: "van"

N.2 "What is sitting in front of the table that looks yellow and black?\nAnswer the question using a single word or phrase."

- **Ground Truth Answer**: "luggage"
- **Vanilla Model**: "backpack"
- **Ours**: "suitcase"

N.3 "What is in front of the poster?\nAnswer the question using a single word or phrase."

- **Ground Truth Answer**: "monitor"
- **Vanilla Model**: "monitor"
- **Ours**: "computer"