CAPA: Contribution-Aware Pruning and FFN Approximation for Efficient Large Vision-Language Models

Anonymous ACL submission

Abstract

Efficient inference in Large Vision-Language Models is constrained by the high cost of processing thousands of visual tokens, yet it remains unclear which tokens and computations can be safely removed. While attention scores are commonly used to estimate visual token importance, they are an imperfect proxy for actual contribution. We show that Attention Contribution, which weights attention probabilities by value vector magnitude, provides a more accurate criterion for visual token selection. Our empirical analysis reveals that visual attention sinks are functionally heterogeneous, comprising Probability Dumps with low contribution that can be safely pruned, and Structural Anchors with high contribution essential for maintaining model performance. Further, we identify substantial redundancy in Feed-Forward Networks (FFNs) associated with visual tokens, particularly in intermediate layers where image tokens exhibit linear behavior. Based on our findings, we introduce CAPA (Contribution-Aware Pruning and FFN Approximation), a dual-strategy framework that prunes visual tokens using attention contribution at critical functional transitions and reduces FFN computation through efficient linear approximations. Experiments on various benchmarks across baselines show that CAPA achieves competent efficiency–performance trade-offs with improved robustness.

1 Introduction

Large Vision Language Models (LVLMs) such as LLaVA (Liu et al., 2023a), Qwen VL (Bai et al., 2025; Wang et al., 2024), and InternVL (Chen et al., 2024b,c; Zhu et al., 2025) have achieved remarkable success by bridging the gap between visual perception and textual reasoning.

However, the computational cost of processing high resolution images, often involving thousands of visual tokens, remains a major bottleneck. While visual token pruning has emerged as a popular solution, most existing methods (Chen et al., 2024a; Endo et al., 2025; Ye et al., 2025; Zhang et al., 2025) rely heavily on attention scores or heuristic metrics derived directly from the attention scores to identify unimportant tokens.

In addition, there have been several works showing that the language model backbones in LVLMs treat image tokens and text tokens separately (Li et al., 2025; Zhang et al., 2024; Jyoti Bajpai and Hanawal, 2025), and hence their redundancy should be handled independently. While prior research in the LLM domain has explored approximating the Feed-Forward Networks (FFNs) through methods such as sparse approximation (Frantar and Alistarh, 2023; Zhang et al., 2022; Liu et al., 2023b; Chavan et al., 2024), these techniques have not been extended to LVLMs, specifically regarding vision tokens. This gap is critical because, as we show further, vision tokens and text tokens do not behave in the same way. Consequently, redundancy in FFNs for vision tokens across layers remains largely unexplored, particularly in identifying which layers exhibit FFN redundancy and which do not.

Through empirical analyses of visual token behaviors in LVLMs, we identify two key observations that existing acceleration methods fail to capture. First, attention scores alone are an insufficient proxy for visual token importance, as they do not account for the magnitude of the associated value vectors, motivating attention contribution as a more faithful signal for visual token pruning. Second, redundancy in LVLMs extends beyond token selection to the FFNs associated with visual tokens, where we observe that, in certain layers, FFN transformations exhibit near-linear behavior.

Building on these observations, we propose CAPA (Contribution-Aware Pruning and FFN Approximation), a dual-strategy framework for efficient LVLM inference. CAPA directly operational-

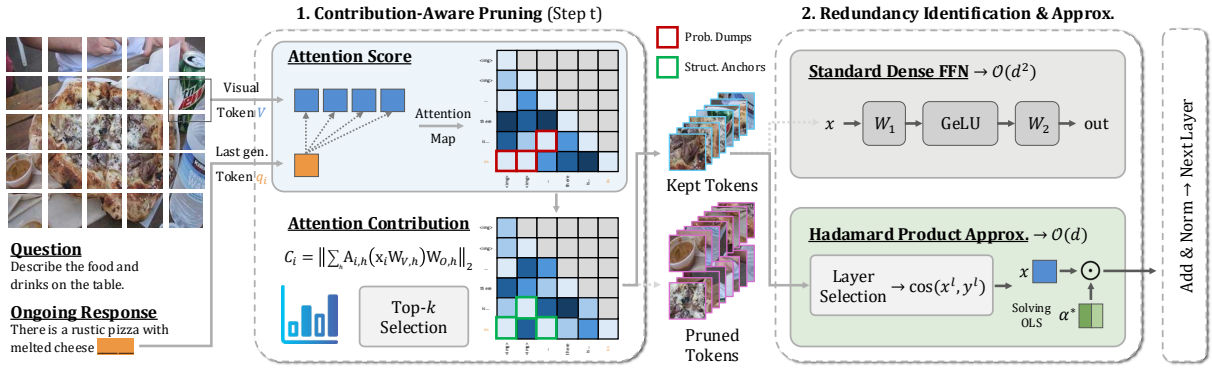

Figure 1: Overall framework of CAPA — (Left) Contribution-Aware Pruning: at each generation step t, we compute the *Attention Contribution* score C_i for every visual token by weighting its value vector magnitude with the attention probability assigned by the last generated token query q_t; (Right) FFN Approximation: in the layers identified as redundant (high input-output cosine similarity), we replace the computationally expensive dense FFNs ($\mathcal{O}(d^2)$) with a lightweight, learned element-wise Hadamard product ($\mathcal{O}(d)$).

izes our empirical findings by addressing redundancy at both the token and computation levels: it prunes visual tokens based on attention contribution rather than raw attention scores, and reduces unnecessary computation by approximating FFN transformations for vision tokens in layers where redundancy is observed. By jointly considering which visual tokens matter and where visual computation can be reduced, CAPA is designed to achieve favorable efficiency–performance trade-offs without compromising model robustness. Extensive experiments across multiple LVLM backbones and benchmarks show that CAPA consistently preserves task performance while delivering substantial inference acceleration, outperforming existing pruning-based baselines.

Our contributions are as follows:

1. We present empirical analyses of visual token processing in LVLMs, showing that attention scores alone are insufficient to capture visual token importance and that significant redundancy exists in FFNs associated with vision tokens across layers.
2. Based on our empirical findings, we propose CAPA, a dual-strategy framework that combines contribution-aware visual token pruning with FFN approximation to reduce inference cost in LVLMs.
3. Through extensive experiments on multiple LVLM backbones and benchmarks, we demonstrate that CAPA achieves strong efficiency–performance trade-offs while maintaining robust model performance.

2 Related Work

Vision Token Pruning in LVLMs. Reducing the computational overhead of Large vision–language models has attracted growing attention, with visual token pruning emerging as a dominant acceleration strategy. Early approaches primarily rely on raw attention scores or heuristic importance measures to discard visually redundant tokens (Chen et al., 2024a; Endo et al., 2025). These methods implicitly assume that attention probability alone reflects token saliency. However, recent analyses have shown that attention weights can be misleading when detached from value representations, particularly in multi-modal settings (Kobayashi et al., 2020; Basu et al., 2024; Guo et al., 2024). In contrast to prior work, we explicitly incorporate value vector magnitude through Attention Contribution, yielding a more faithful estimate of a token's functional impact on downstream representations and enabling more reliable visual token pruning.

Visual Attention Sinks. Several studies have observed the emergence of attention sinks particularly visual attention sinks in LVLMs, where certain tokens attract disproportionately high attention mass (Kang et al., 2025; Queipo-de Llano et al., 2025; Xiao et al., 2023). These works typically treat attention sinks as a single phenomenon and focus on mitigating their negative effects through masking or reweighting. Our work departs from this view by demonstrating that visual attention sinks are functionally heterogeneous. Using attention contribution, we distinguish between low-contribution probability dumps and high-contribution structural anchors, enabling safe pruning without compromis-

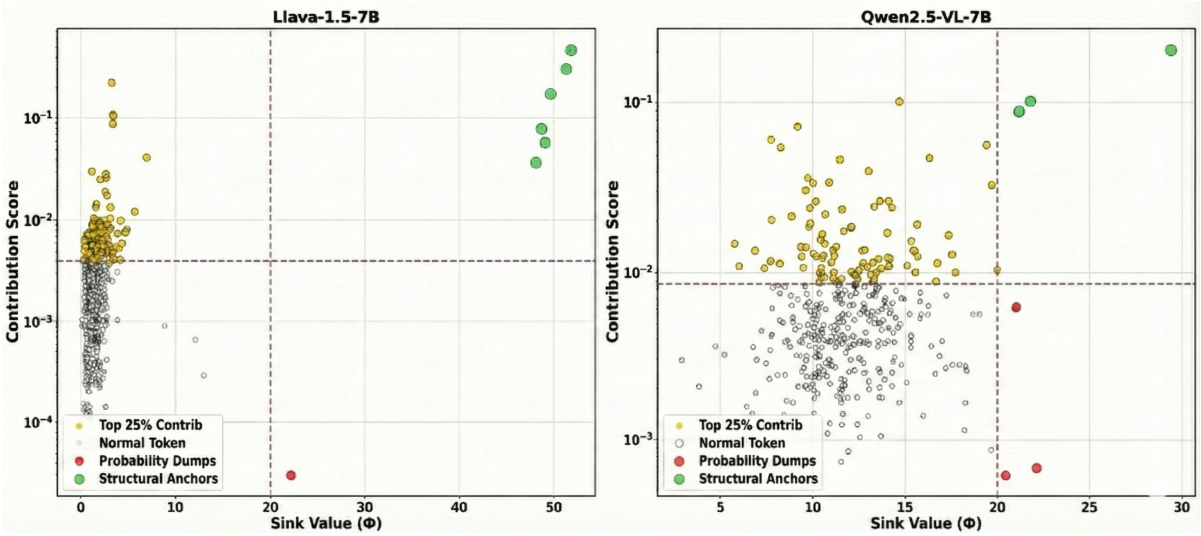

Figure 2: Sink distribution analysis across LLaVA-1.5 and Qwen2.5-VL. Using a sink threshold of $\tau = 20$, we classify tokens into two groups: low-contribution **Probability Dumps** (Type I) and high-contribution **Structural Anchors** (Type II) . The dashed line represents the sink identification threshold.

ing model integrity.

Feed Forward Network Approx. and Efficiency. The FFNs constitute approximately two-thirds of the parameter count in Transformer models, making them a prime target for optimization. In the realm of LLMs, extensive research has attempted to address FFN redundancy by replacing standard dense layers with significantly more complex formulations. These include substituting FFNs with Mixture-of-Experts (MoE) architectures to route tokens to specialized expert blocks (Zhang et al., 2022; Gale et al.), replacing dense matrices with structured linear parameterizations or Monarch matrices (Fu et al., 2023; Wei et al., 2024). Other works have explored even more radical structural changes, such as approximating FFNs with Kolmogorov-Arnold Networks (KAN) using learnable spline functions (Liu et al., 2024b), deploying spiking neural networks for event-driven sparsity (Zhu et al., 2023), or applying tensor decomposition like Tucker or CP decomposition to compress weight matrices (Wang et al.; Xu et al., 2023).

To the best of our knowledge no comparable work has investigated FFN redundancy specifically for vision tokens in LVLMs. We bridge this gap by identifying that visual tokens exhibit strong linearity in intermediate layers distinct from text tokens allowing us to replace expensive FFNs with a lightweight, learned Hadamard product rather than complex architectural substitutes.

3 Empirical Analysis

3.1 Deconstructing Visual Attention Sinks

To revisit the functional role of high-attention visual tokens, we first examine the extent to which statistical presence (high attention weights) aligns with representational impact (contribution to the residual stream). While prior work (Kang et al., 2025) identifies *visual attention sinks* solely based on massive activation in specific hidden dimensions (defined as Sink Value $\phi > \tau$ with $\tau = 20$), we observe that this definition may conflate functionally distinct token behaviors. Accordingly, we analyze the token distribution of LLaVA-1.5 (Liu et al., 2023a) and Qwen2.5-VL (Bai et al., 2025) by correlating two orthogonal metrics for every visual token: the Sink Value (ϕ), representing the activation magnitude in outlier dimensions, and the Attention Contribution (C_i), representing the weighted value vector's magnitude added to the residual stream (defined in Sec 4.1).

As visualized in Fig. 2, this multi-dimensional analysis reveals a critical functional dichotomy among tokens that arguably qualify as *sinks* under the standard definition ($\phi > 20$). We observe that visual tokens do not form a monolithic group; rather, they bifurcate into two distinct clusters based on their contribution scores:

- **Type I: Probability Dumps** are clustered in the low-contribution region (highlighted in red), these tokens exhibit the classic behavior described in recent literature. Despite possess-

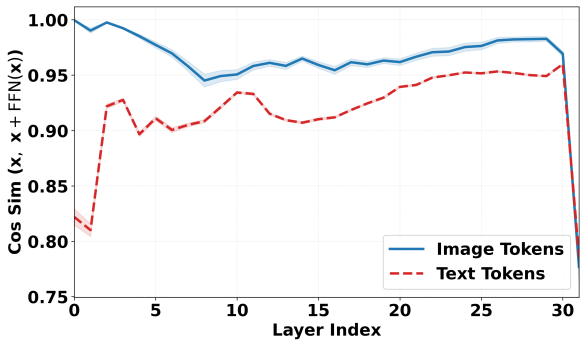

Figure 3: Layer-wise FFN Redundancy for Image vs. Text Tokens. The plot displays the mean cosine sim. $\cos(\mathbf{x}, \mathbf{x} + \text{FFN}(\mathbf{x}))$ across network layers, evaluated on 500 samples from the MSCOCO 2017 val set.

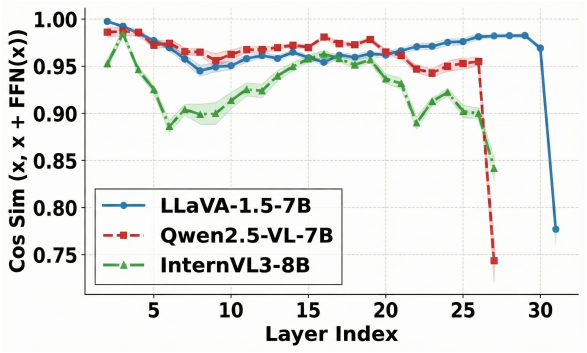

Figure 4: Comparative analysis of FFN redundancy across LVLM architectures. We plot the average cosine sim. between the input hidden state $\mathbf{x}$ and the residual output $\mathbf{x} + \text{FFN}(\mathbf{x})$ for three baselines.

ing high attention weights and high sink values ($\phi > 20$), their resulting contribution C_i is negligible. This confirms that their primary function is to serve as passive receptacles for excess probability mass generated by the Softmax operation, rather than encoding semantic content. Consequently, they represent true redundancy and can be pruned safely.

- **Type II: Structural Anchors** are, crucially, identified as a second cluster (highlighted in green) that standard sink thresholds only fail to distinguish. They couple high sink values with massive value vector magnitudes, resulting in high Attention Contribution (C_i). Unlike probability dumps, these tokens act as critical biases or *anchors* within the residual stream. Our analysis shows that while they statistically resemble sinks due to high ϕ, their removal leads to immediate representational collapse.

This empirical distinction demonstrates that attention scores alone are insufficient for pruning while maintaining visual perception capabilities.

3.2 Modality-Dependent FFN Redundancy

After re-examining the functional role of visual tokens at the attention level, we now turn to their role in FFNs, which constitute massive parameter count in Transformers and represent a primary computational bottleneck during inference. While prior research in LLMs (Pessoa Pires et al., 2023; Gromov et al., 2024; Bercovich et al., 2025) has explored FFN redundancy, these works typically treat all tokens uniformly. In contrast, we hypothesize that in LVLMs, the language backbone processes visual and text tokens with different degrees of non-linearity. To delve into this, we design an experiment to quantify the *functional necessity* of FFN layers specifically for visual tokens.

We analyze the linearity of FFN transformations by measuring the directional alignment between the layer input $\mathbf{x}$ and the post-FFN residual output $\mathbf{y} = \mathbf{x} + \text{FFN}(\mathbf{x})$. Intuitively, if an FFN contributes significant non-linear processing, the output vector should diverge effectively from the residual connection. Conversely, a cosine similarity near 1.0 implies an identity-like mapping where the FFN is redundant. We formalize this metric as:

$$\text{Sim}(\mathbf{x}) = \cos(\mathbf{x}, \mathbf{x} + \text{FFN}(\mathbf{x})). \quad (1)$$

We evaluated this metric across all layers of LLaVA-1.5-7B, Qwen2.5-VL-7B, and InternVL3-8B using 500 samples from the MSCOCO 2017 validation set (Lin et al., 2014).

Our layer-wise analysis, as visualized in Fig. 3, reveals a striking disparity in processing patterns between modalities. Image tokens (solid blue line) consistently maintain near-identity similarity ($>$ 0.96) across the majority of layers. In contrast, text tokens (dashed red line) exhibit lower similarity in early and middle layers, indicating a higher reliance on FFN transformations, before converging to high redundancy in the deeper layers. Both modalities show a significant drop in similarity at the final layer. This distinction is not unique to a single architecture; as demonstrated in Fig. 4, this pattern of high visual token linearity in intermediate layers is consistent across three baselines.

This structural redundancy suggests that for visual tokens, the expensive FFN computations in these layers effectively collapse into linear transformations. This indicates that the model's depth is not fully utilized for visual processing, motivating our proposed strategy to replace these dense

layers with efficient linear approximations without compromising representational integrity.

4 Methodology

Taken together, our empirical analyses show that inefficiency in LVLM inference arises from complementary sources. At the attention level, visual tokens with high attention mass exhibit heterogeneous functional roles, rendering attention score pruning unreliable. At the computation level, FFN transformations for visual tokens exhibit substantial linear redundancy. These findings suggest that effective acceleration requires jointly reasoning about which visual tokens matter and where visual computation can be simplified, motivating CAPA, a dual-strategy framework that integrates contribution-aware visual token pruning with FFN approximation as summarized in Fig. 1.

4.1 Contribution-Aware Pruning

To effectively identify redundant visual tokens, it is crucial to distinguish between the probability of attending to a token and the actual informational impact of that token on the network's processing. Standard pruning methods have solely exploited attention scores, implicitly assuming that higher attention weights correlate directly with feature importance. However, attention weights function primarily as routing coefficients—they determine which tokens are selected, but not necessarily the magnitude of the update applied to the residual stream. The actual quantity of information transferred depends on the Value vectors projected by the attention heads. A token with a high attention score but a negligible value vector magnitude contributes little to the representation.

To address it, we quantify the importance of a visual token i using its attention contribution score C_i, which integrates both the routing weight and the feature magnitude. Specifically, we compute the weighted aggregation of the value vectors projected by the output matrix $\mathbf{W}_{O,h}$ to capture the total magnitude of information transfer. It is defined as the ℓ_2 norm of this contribution to the residual stream:

$$C_i = \left\| \sum_{h=1}^{H} \mathbf{A}_{i,h} \big(\mathbf{x}_i \mathbf{W}_{V,h}\big) \mathbf{W}_{O,h} \right\|_2, \qquad (2)$$

where $\mathbf{A}_{i,h}$ denotes the attention probability assigned to token i by head h, $\mathbf{x}_i$ is the input visual token representation, and $\mathbf{W}_{V,h}$, $\mathbf{W}_{O,h}$ are the value and output projection matrices. By measuring the norm of the projected vector, this formulation captures the effective contribution of each token to the residual update, providing a more faithful proxy for token saliency than raw attention weights alone.

In practice, we apply this metric dynamically during the generation process. At every generation step, we calculate C_i relative to the current query token (the last generated token). We then select the top-k visual tokens with the highest contribution scores to be retained in the key-value cache, while tokens with minimal contribution are pruned. This ensures that the retained visual context is not static; instead, it dynamically adapts to the specific semantic requirements of the current generation step, preserving only the visual information that functionally alters the model's output distribution.

4.2 Redundancy Identification & Approx.

Having addressed the redundancy in the sequence length (N) through token pruning, we essentially employ a dual-strategy approach by next targeting the redundancy in the model's width d. As in Fig. 1, we delve into FFNs, which account for the majority of the remaining FLOPs, to estimate their necessity and propose a lightweight approximation.

Layer Selection Strategy. To identify layers suitable for approximation, we analyze the transformation magnitude of the FFNs across the model. We define the redundancy of a layer l by the cosine similarity between its input hidden state $\mathbf{x}^{(l)}$ and the output of the residual block $\mathbf{y}^{(l)} = \mathbf{x}^{(l)} + \text{FFN}(\mathbf{x}^{(l)})$. As we observed in Fig. 4, layers exhibiting high cosine similarity imply a near-identity transformation, where the FFN contributes minimal non-linear modification to the feature space.

We formalize our selection criterion $\mathcal{S}$ with a threshold η. A layer is selected for approximation if the similarity over the calibration set exceeds η:

$$\mathcal{S} = \{l \mid \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \left[\cos(\mathbf{x}^{(l)}, \mathbf{y}^{(l)})\right] > \eta\} \qquad (3)$$

Based on this analysis, we target layers that fall within high-redundancy regions while preserving critical processing stages (*e.g.,* final output layers).

Optimized Hadamard Product Approximation. For layers $l \in \mathcal{S}$, we replace the computationally expensive matrix operations of the FFN with a learned, lightweight element-wise scaling vector $\boldsymbol{\alpha} \in \mathbb{R}^d$. This collapses the entire residual FFN

Table 1: Pruning performance comparison of CAPA against baselines *L*: LLaVA-1.5-7B, *Q*: Qwen2.5-VL-7B, and *I*: InternVL3-8B. CAPA demonstrates consistent robustness and superior performance across all benchmarks.

l	Method	VQA2			MMBench			MMVet			TextVQA			SEED			MMMU		
		L	*Q*	*I*	*L*	*Q*	*I*	*L*	*Q*	*I*	*L*	*Q*	*I*	*L*	*Q*	*I*	*L*	*Q*	*I*
Full	Vanilla	76.55	83.74	80.28	62.97	82.21	84.71	28.89	61.83	70.41	48.64	83.69	80.89	60.14	76.80	74.97	34.67	48.68	55.44
Early	Unif.	72.79	82.61	79.56	61.25	74.10	82.65	24.03	59.31	67.56	32.98	80.82	81.34	56.87	71.20	73.92	34.62	47.10	54.67
	FastV	72.14	82.73	78.26	62.37	75.34	82.71	24.67	60.91	68.80	42.56	82.24	78.87	56.01	73.80	72.02	34.56	47.67	53.42
	Feather	73.87	82.71	79.26	62.45	75.10	82.70	26.30	60.68	69.95	45.30	81.24	79.87	56.86	73.10	73.02	34.70	47.67	54.43
	CAPA	**74.76**	**83.10**	**80.26**	**62.80**	**78.16**	**84.79**	**28.75**	**62.51**	**70.37**	**46.70**	**82.90**	**80.92**	**57.91**	**75.30**	**74.02**	**35.44**	**47.97**	**55.44**
Transition	Unif.	73.95	82.61	79.05	62.60	75.20	84.62	25.69	62.11	68.34	36.07	81.74	80.65	58.80	73.92	73.00	35.44	47.67	55.00
	FastV	75.27	82.75	79.50	62.43	75.93	83.10	27.01	61.42	69.31	47.05	81.55	79.10	56.91	73.65	71.27	35.56	47.69	54.27
	Feather	75.28	82.79	79.56	62.54	75.30	83.20	26.91	61.83	69.95	47.46	82.10	80.12	57.05	73.69	72.02	35.67	47.67	53.21
	CAPA	**76.21**	**84.10**	**80.26**	**62.82**	**77.20**	**84.79**	**27.96**	**64.25**	**69.85**	**48.40**	**82.34**	**81.20**	**59.94**	**75.72**	**74.02**	**36.61**	**47.89**	**55.44**
Late	Unif.	74.65	82.62	79.90	62.47	75.10	83.00	25.55	59.95	69.49	37.87	81.78	80.35	59.17	71.70	73.02	34.22	48.44	51.40
	FastV	75.27	82.75	80.10	62.43	76.40	84.30	28.27	61.74	69.27	47.38	82.37	80.98	59.13	73.75	72.50	35.67	48.54	51.60
	Feather	75.52	82.80	80.30	62.43	76.70	84.70	28.48	63.30	69.35	47.43	82.39	80.98	59.13	73.79	71.79	35.68	48.59	51.67
	CAPA	**76.20**	**84.30**	**80.27**	**62.67**	**77.70**	**84.79**	**28.70**	**64.70**	**69.63**	**48.38**	**82.30**	**81.93**	**60.08**	**75.60**	**72.02**	**35.89**	**49.33**	**53.67**

block into a single Hadamard product operation:

$$\mathbf{y}^{(l)} \approx \hat{\mathbf{y}}^{(l)} = \mathbf{x}^{(l)} \odot \boldsymbol{\alpha} \tag{4}$$

where $\odot$ denotes the element-wise product. It reduces the layer's FLOPs count from $\mathcal{O}(d^2)$ to $\mathcal{O}(d)$. More discussion on the complexity in A.1

Closed-Form Least Squares Solution. To find the optimal $\boldsymbol{\alpha}$, we avoid iterative gradient descent and instead solve the Ordinary Least Squares (OLS) objective analytically. We collect activation statistics using 500 samples from the COCO 2017 train set (Lin et al., 2014). For each feature dimension $k \in \{1, \ldots, d\}$, we minimize the squared reconstruction error between the approximated output and the true residual output over N calibration samples:

$$\min_{\alpha_k} \sum_{n=1}^{N} \left(\alpha_k x_{n,k} - y_{n,k}\right)^2. \tag{5}$$

Setting the partial derivative with respect to α_k to zero yields a closed-form solution based on the second-order moments of the features:

$$\alpha_k^* = \frac{\sum_{n=1}^{N} x_{n,k} \cdot y_{n,k}}{\sum_{n=1}^{N} x_{n,k}^2}. \tag{6}$$

In vector notation, this is computed as $\boldsymbol{\alpha}^* = (\sum \mathbf{x}_n \odot \mathbf{y}_n) \oslash (\sum \mathbf{x}_n \odot \mathbf{x}_n)$, where $\oslash$ represents element-wise division. This approach ensures $\boldsymbol{\alpha}$ captures the optimal linear scaling factor that statistically minimizes error across the data distribution. We observe that feature statistics converge rapidly, making 500 samples sufficient for a robust approximation without extensive retraining.

5 Experiments

5.1 Experimental Setup

Baseline Models and Benchmarks. We evaluate our framework on three state-of-the-art LVLMs: LLaVA-1.5-7B (Liu et al., 2023a), Qwen2.5-VL-7B (Bai et al., 2025), and InternVL3-8B (Zhu et al., 2025). To ensure a comprehensive assessment of multi-modal capabilities, we report results on six standard benchmarks: VQAv2 (Goyal et al., 2017) for general visual QA, MMBench (Liu et al., 2024a) and SEED-Bench (Li et al., 2023) for holistic perception, MM-Vet (Yu et al., 2023) for integrated reasoning, TextVQA (Singh et al., 2019) for OCR capabilities, and MMMU (Yue et al., 2024) for multi-discipline expert reasoning.

Pruning Baselines. We compare CAPA against a various set of pruning strategies to validate its efficacy. As foundational baselines, we include the **Vanilla** (unpruned) model and a **Uniform** baseline, which naively samples tokens with a fixed stride to match the target token budget. Among attention-centric methods, we evaluate **FastV** (Chen et al., 2024a), a training-free approach that prunes tokens based on raw attention scores in early layers, and **Feather** (Endo et al., 2025), which modifies causal masking constraints. For Feather, we implement the method without its ensemble mechanism to isolate the efficacy of its *No-RoPE* heuristic for fair comparison.

Implementation Details. For all pruning experiments, we retain 25% of the visual tokens. Following prior observations (Luo et al., 2025; Yu and Lee, 2025) that multi-modal transformers exhibit depth-dependent behavior, we partition layers into three stages: *early*, *transition*, and *late*. Pruning is applied at the phase-transition layers identified in (Yu and Lee, 2025) (LLaVA: Layers 5, 12, 16; QwenVL: Layers 3, 11, 21; InternVL: Layers 3, 11, 21). For FFN approximation, we target the high-redundancy blocks as highlighted in Sec. 4.2 and

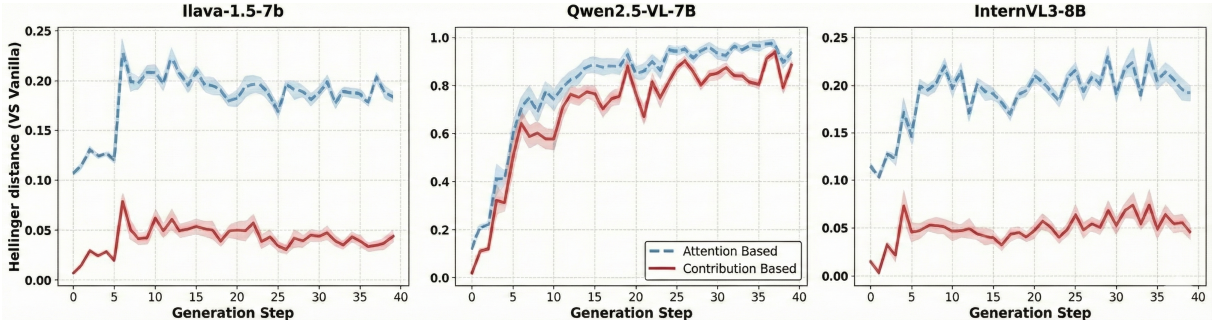

Figure 5: Hellinger distance between the vanilla model and pruned logits across generation steps, comparing attention-score and attention-contribution pruning for 3 baselines on the COCO2017 val. set (200 examples).

Table 2: Impact of different FFN replacement strategies applied to both the Vanilla (unpruned) and CAPA (pruned) models. The Hadamard product recovers most of the performance degradation caused by skipping FFNs. *L*: LLaVA-1.5-7B, *Q*: Qwen2.5-VL-7B, and *I*: InternVL3-8B.

Type	Strategy	VQA2			MMBench			MMVet			TextVQA			SEED			MMMU		
		L	*Q*	*I*	*L*	*Q*	*I*	*L*	*Q*	*I*	*L*	*Q*	*I*	*L*	*Q*	*I*	*L*	*Q*	*I*
Vanilla	Full Model	76.55	83.74	80.28	62.97	82.21	84.71	28.89	61.83	70.41	48.64	83.69	80.89	60.14	76.80	74.97	34.67	48.68	55.44
	Skip FFN	74.02	79.15	75.70	58.12	72.40	78.13	22.45	49.30	59.10	44.10	76.12	74.90	56.78	69.20	67.32	29.30	38.50	42.37
	Hadamard	**75.48**	**81.02**	**78.79**	**61.32**	**77.85**	**80.52**	**26.90**	**56.40**	**65.05**	**47.15**	**80.22**	**78.03**	**58.82**	**73.15**	**71.95**	**32.85**	**44.10**	**49.33**
CAPA	Prune Only	76.20	84.30	80.27	62.67	77.70	84.79	28.70	64.70	69.63	48.38	82.30	81.93	60.08	75.60	72.02	35.89	49.33	53.67
	Skip FFN	73.88	79.60	73.90	57.90	67.20	75.80	22.10	52.15	56.45	43.85	74.50	73.12	56.40	67.90	64.50	30.25	39.45	40.85
	Hadamard	**75.10**	**81.85**	**76.70**	**61.05**	**73.20**	**78.34**	**26.44**	**59.12**	**62.30**	**46.90**	**78.90**	**75.08**	**58.65**	**71.42**	**71.30**	**34.02**	**45.22**	**45.33**

we keep η as 0.96 and hence find the redundant layers (LLaVA: Layers 2–5 & 22–29; Qwen: Layers 2–5 & 13–19 ; InternVL Layers 2–4 & 14–20). The approximation parameters α are calibrated using 500 randomly sampled images from the COCO train set (Lin et al., 2014).

5.2 Main Results: Pruning Performance

Tab. 1 reports layer-wise pruning results for three baselines, with evaluations conducted at critical *phase transition* layers to assess robustness under representational shifts. Across all models, pruning robustness consistently improves in later layers, indicating that dependence on dense visual tokens diminishes as representations mature, rendering image tokens increasingly redundant in deeper layers.

CAPA demonstrates a clear advantage at the *transition* layers (Fig. 6), which represent the most sensitive stage for token reduction. Baseline methods such as FastV and Feather show substantial performance degradation in this phase, whereas CAPA maintains performance close to the unpruned Vanilla model. Task-level analysis further shows that CAPA is particularly effective on benchmarks requiring fine-grained perception and complex reasoning (TextVQA, MMMU). This suggests that semantically dense tokens, such as image-embedded text or intricate visual details, often exhibit high value vector magnitudes despite variable attention scores. By explicitly weighting these

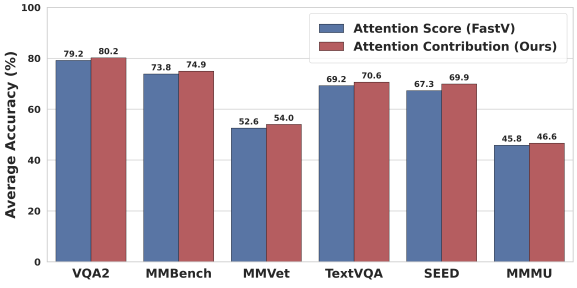

Figure 6: Performance comparison of standard Attention Score pruning (FastV) against our Attention Contribution method (CAPA) at the transition layers. Results are averaged across 3 baselines for generalizability.

contributions, CAPA avoids removing information-dense tokens, a common failure mode of attention-only or uniform pruning. Finally, pruning in *late* layers is universally safe, with all methods converging to near-Vanilla performance, validating the high redundancy of visual tokens in the final layers.

5.3 Ablation Study for CAPA

Logit Divergence under Attention Pruning. To compare how different attention-based pruning criteria preserve the generative behavior of the vanilla model, we measure the divergence between vanilla models and their pruned variants during long-form generation. All tokens before the pruning step are kept identical, ensuring that any deviation is solely induced by the choice of pruning criterion.

Specifically, let x denote the visual input, $\tilde{x}$ the pruned visual input, and $y_{<t}$ the generated pre-

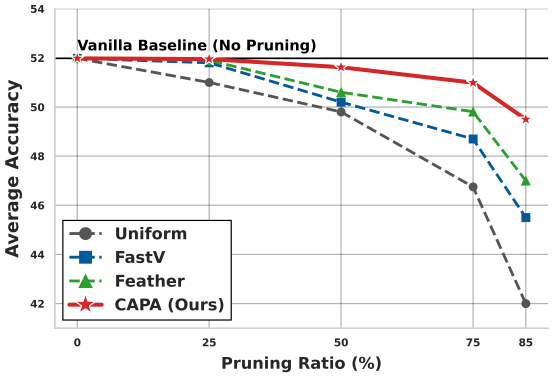

Figure 7: Average accuracy across six benchmarks under varying pruning ratios. While baselines exhibit rapid degradation at high sparsity levels, **CAPA** maintains near-lossless performance up to 85% sparsity.

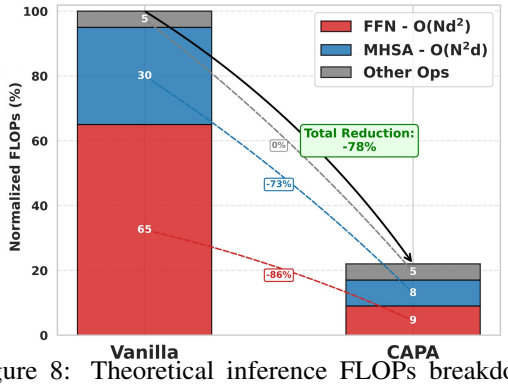

Figure 8: Theoretical inference FLOPs breakdown across model components. By identifying and approximating redundant FFNs (red), CAPA effectively removes the dominant $\mathcal{O}(Nd^2)$ cost with 78% reduction.

fix up to step t. We compare the output distributions $p(\cdot \mid y_{<t}, x)$ from the vanilla model and $\tilde{p}(\cdot \mid y_{<t}, \tilde{x})$ from the pruned model, where pruning is performed using either attention scores or attention contribution.

We quantify the discrepancy between these distributions using Hellinger distance (Le Cam, 2012):

$$H(p, \tilde{p}) = \frac{1}{\sqrt{2}} \left(\sum_i \left(\sqrt{p_i} - \sqrt{\tilde{p}_i} \right)^2 \right)^{1/2}, \quad (7)$$

where p_i and $\tilde{p}_i$ denote the probabilities assigned to token i by vanilla and pruned models, respectively.

As shown in Fig. 5, pruning based on attention contribution consistently yields lower Hellinger distances than pruning based on raw attention scores across all evaluated models. This gap widens at later generation steps, indicating that attention contribution more faithfully preserves the vanilla model's output distribution over long-horizon multi-modal generation.

FFN Approximation Efficiency. Tab. 2 delineates the impact of our approximation strategies, both in isolation and when coupled with token pruning. We first observe that naively removing FFNs (*Vanilla + Skip FFN*) leads to significant performance degradation, confirming that these layers retain non-negligible functional importance. However, replacing them with our proposed linear approximation (*Vanilla + Hadamard product*) successfully recovers the majority of this performance drop, validating our hypothesis that the transformation in these layers is predominantly linear. When integrated with contribution-aware pruning, simply skipping FFN layers (*CAPA with Skip FFN*) proves insufficient; in contrast, the fully combined framework (*CAPA Combined*) which pairs token pruning with the Hadamard product approximation achieves the optimal trade-off between computational efficiency and model accuracy.

5.4 Robustness and Efficiency Analysis

We further analyze the trade-off between computational efficiency and model performance. Fig. 7 illustrates the robustness of CAPA across varying pruning ratios. Unlike prior methods such as FastV (Chen et al., 2024a) and Feather (Endo et al., 2025), which experience precipitous performance drops beyond 50% pruning, CAPA maintains near-vanilla accuracy (≈ 52.0 average score) even at an aggressive 75% pruning ratio. This stability validates that Attention Contribution is a superior proxy for token saliency than raw attention scores.

In addition, we quantify the computational gains in Fig. 8. Standard LVLM inference is dominated by FFNs, which account for approximately 65% of total FLOPs. By replacing these dense matrix multiplications with our lightweight approximation in redundant layers, CAPA virtually eliminates this bottleneck (red region). When combined with the quadratic speedup from token pruning (blue region), our framework achieves a total FLOPs reduction of 78%, offering a decisive efficiency over baselines that rely on token pruning alone.

6 Conclusion

In this paper, we introduced CAPA, an efficient framework for LVLM inference. By shifting pruning from attention scores to attention contribution magnitude and approximating redundant FFNs via lightweight Hadamard products, CAPA preserves critical visual information while significantly reducing computation. Our results highlight the need for visual-specific optimization distinct from text to maximize efficiency without sacrificing capability.

7 Limitations

While our post-hoc strategies contribution-aware pruning and FFN approximation yield substantial efficiency gains, they are not a complete solution. Post-hoc methods operate on fixed, pretrained backbones and therefore cannot fully exploit the benefits of jointly learned token selection and model computation. A promising direction for future work is the design of learned controllers that adaptively decide, for each input and at each layer, how many visual tokens to retain (or how much computation to allocate). Such layer-wise and data-dependent policies could enable finer-grained trade-offs between computation and accuracy than static pruning heuristics.

Another important avenue is architectural: developing vision–language backbones in which the standard `FFN` is replaced or reparameterized by more efficient modules that are specifically tailored to visual tokens (*e.g.,* lightweight element-wise transforms, low-rank or conditional linear layers, or mixture-of-expert style blocks). Jointly optimizing token-retention policies and such efficient FFN alternatives during training may yield models that are both faster and more robust than what post-hoc modifications can achieve.

Finally, future studies should evaluate these ideas across diverse benchmarks and distribution shifts, investigate calibration and interpretability of learned policies, and quantify the trade-offs between dynamic sparsity, latency, and downstream task performance.

References

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*.

Samyadeep Basu, Martin Grayson, Cecily Morrison, Besmira Nushi, Soheil Feizi, and Daniela Massiceti. 2024. Understanding information storage and transfer in multi-modal large language models. *Advances in Neural Information Processing Systems*, 37:7400–7426.

Akhiad Bercovich, Mohammad Dabbah, Omri Puny, Ido Galil, Amnon Geifman, Yonatan Geifman, Izhak Golan, Ehud Karpas, Itay Levy, Zach Moshe, and 1 others. 2025. Ffn fusion: Rethinking sequential computation in large language models. *arXiv preprint arXiv:2503.18908*.

Arnav Chavan, Nahush Lele, and Deepak Gupta. 2024. Surgical feature-space decomposition of llms: Why, when and how? *arXiv preprint arXiv:2405.13039*.

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. In *European Conference on Computer Vision*, pages 19–35. Springer.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024b. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024c. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198.

Mark Endo, Xiaohan Wang, and Serena Yeung-Levy. 2025. Feather the throttle: Revisiting visual token pruning for vision-language model acceleration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22826–22835.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pages 10323–10337. PMLR.

Dan Fu, Simran Arora, Jessica Grogan, Isys Johnson, Evan Sabri Eyuboglu, Armin Thomas, Benjamin Spector, Michael Poli, Atri Rudra, and Christopher Ré. 2023. Monarch mixer: A simple sub-quadratic gemm-based architecture. *Advances in Neural Information Processing Systems*, 36:77546–77603.

Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. Megablocks: Efficient sparse training with mixture-of-experts, 2022. *URL https://arxiv.org/abs/2211.15841*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. 2024. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*.

Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. 2024. Attention score is not all you need for token importance indicator in kv cache reduction: Value also matters. *arXiv preprint arXiv:2406.12335*.

Divya Jyoti Bajpai and Manjesh Kumar Hanawal. 2025. Free: Fast and robust vision language models with early exits. *arXiv e-prints*, pages arXiv–2506.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. 2025. See what you are told: Visual attention sink in large multimodal models. *arXiv preprint arXiv:2503.03321*.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. *arXiv preprint arXiv:2004.10102*.

Lucien Le Cam. 2012. *Asymptotic Methods in Statistical Decision Theory*. Springer.

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

Hongliang Li, Jiaxin Zhang, Wenhui Liao, Dezhi Peng, Kai Ding, and Lianwen Jin. 2025. Redundancylens: Revealing and exploiting visual token processing redundancy for efficient decoder-only mllms. *arXiv preprint arXiv:2501.19036*.

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 *European conference on computer vision*, pages 740–755. Springer.

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

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024a. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and 1 others. 2023b. Deja vu: Contextual sparsity for efficient llms at inference time. In *International Conference on Machine Learning*, pages 22137–22176. PMLR.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. 2024b. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*.

Xuan Luo, Weizhi Wang, and Xifeng Yan. 2025. Direct multi-token decoding. *arXiv preprint arXiv:2510.11958*.

Telmo Pessoa Pires, António V Lopes, Yannick Assogba, and Hendra Setiawan. 2023. One wide feedforward is all you need. *arXiv e-prints*, pages arXiv–2309.

Enrique Queipo-de Llano, Álvaro Arroyo, Federico Barbero, Xiaowen Dong, Michael Bronstein, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Attention sinks and compression valleys in llms are two sides of the same coin. *arXiv preprint arXiv:2510.06477*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression, 2024. *URL https://arxiv. org/abs/2403.07378*.

Xiuying Wei, Skander Moalla, Razvan Pascanu, and Caglar Gulcehre. 2024. Building on efficient foundations: Effective training of llms with structured feedforward layers. *Advances in Neural Information Processing Systems*, 37:4689–4717.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Mingxue Xu, Yao Lei Xu, and Danilo P Mandic. 2023. Tensorgpt: Efficient compression of large language models based on tensor-train decomposition. *arXiv preprint arXiv:2307.00526*.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. 2025. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22128–22136.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

Zhuoran Yu and Yong Jae Lee. 2025. How multimodal llms solve image tasks: A lens on visual grounding, task reasoning, and answer decoding. *arXiv preprint arXiv:2508.20279*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. 2025. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. *arXiv preprint arXiv:2412.01818*.

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

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. Moefication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 877–890.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

Rui-Jie Zhu, Qihang Zhao, Guoqi Li, and Jason K Eshraghian. 2023. Spikegpt: Generative pre-trained language model with spiking neural networks. *arXiv preprint arXiv:2302.13939*.

A Appendix

A.1 Computational Complexity Analysis

In this section, we provide a theoretical analysis of the computational efficiency gains introduced by our proposed framework. We examine the Floating Point Operations (FLOPs) reduction achieved through two primary mechanisms: (1) Hadamard Product Approximation for redundant FFN layers, and (2) Contribution-Aware Vision Token Pruning.

A.1.1 Efficiency of Hadamard FFN Approximation

Standard LVLMs such as LLaVA and QwenVL utilize SwiGLU-based Feed-Forward Networks (FFNs). Let d denote the hidden dimension size and d_{ff} denote the intermediate dimension size (typically $d_{ff} \approx 4d$ or higher).

Baseline Complexity (Standard FFN). A standard SwiGLU FFN consists of three dense matrix multiplications (Gate, Up, and Down projections). For a single token, the computational cost is:

$$\text{FLOPs}_{\text{FFN}} = 2 \cdot (3 \cdot d \cdot d_{ff}) = 6dd_{ff} \quad (8)$$

where the factor of 2 accounts for the multiply-accumulate operations.

Approximated Complexity (Hadamard Product). Our proposed approximation replaces these matrix operations with a single element-wise Hadamard product $\mathbf{x} \odot \boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathbf{R}^d$. The cost for this operation is:

$$\text{FLOPs}_{\text{Had}} = d \quad (9)$$

Reduction Analysis. The reduction factor $\mathcal{R}$ for a single approximated layer is:

$$\mathcal{R} = \frac{\text{FLOPs}_{\text{FFN}}}{\text{FLOPs}_{\text{Had}}} = \frac{6dd_{ff}}{d} = 6d_{ff} \quad (10)$$

Given that d_{ff} is in the order of thousands (e.g., $d_{ff} = 11,008$ for LLaMA-2-7B), the computational cost of the approximated layer becomes negligible ($\approx 0.01\%$ of the original cost). For a model with L total layers where L_{approx} layers are selected for approximation, the total FFN FLOPs reduction is proportional to $\frac{L_{approx}}{L}$.

A.1.2 Impact of Contribution-Aware Token Pruning

We reduce the visual token sequence length by pruning 75% of tokens based on their attention contribution scores. Let N_{img} be the number of image tokens and N_{txt} be the number of text tokens. The total sequence length is $N = N_{img} + N_{txt}$. Let $\rho = 0.75$ represent the pruning rate applied to image tokens. The reduced sequence length is effective for all layers subsequent to the pruning stage. The effective sequence length becomes:

$$N' = (1 - \rho)N_{img} + N_{txt} \quad (11)$$

Reduction in Linear Projections (FFNs and QKV). Linear layers (FFNs and Attention projections) have a complexity of $\mathcal{O}(N \cdot d^2)$. The FLOPs reduction is linear with respect to the token count:

$$\text{Speedup}_{\text{Linear}} = \frac{N}{N'} \approx \frac{1}{1-\rho} \quad (N_{\text{img}} \gg N_{\text{txt}}). \quad (12)$$

For $\rho = 0.75$, this yields a theoretical **4× reduction** in FLOPs for all dense layers operating on the visual sequence.

Reduction in Attention Mechanism. The self-attention mechanism has a complexity of $\mathcal{O}(N^2 \cdot d)$. Since the complexity is quadratic with respect to sequence length, the pruning yields a significantly higher speedup:

$$\text{Speedup}_{\text{Attn}} = \left(\frac{N}{N'}\right)^2 \approx \frac{1}{(1-\rho)^2} \quad (13)$$

With $\rho = 0.75$, the attention mechanism theoretically becomes **16× faster** for the visual component.

Total Theoretical Reduction. Combining both strategies, our framework achieves a compounding efficiency gain. The Hadamard product approximation eliminates the $\mathcal{O}(d^2)$ cost of FFNs in redundant layers entirely, while token pruning reduces the N coefficient for all remaining active layers. This dual approach ensures that we attack the computational bottleneck from both the *width* (hidden dimension complexity) and the *length* (sequence complexity) of the model.

A.2 Derivation of Optimal Hadamard Scaling

In this section, we provide the complete mathematical derivation for the closed-form solution used to approximate the Feed-Forward Network (FFN) layers. As described in Section 4.2, our objective is to replace the computationally expensive dense matrix operations in redundant layers with a lightweight, element-wise Hadamard product. We achieve this

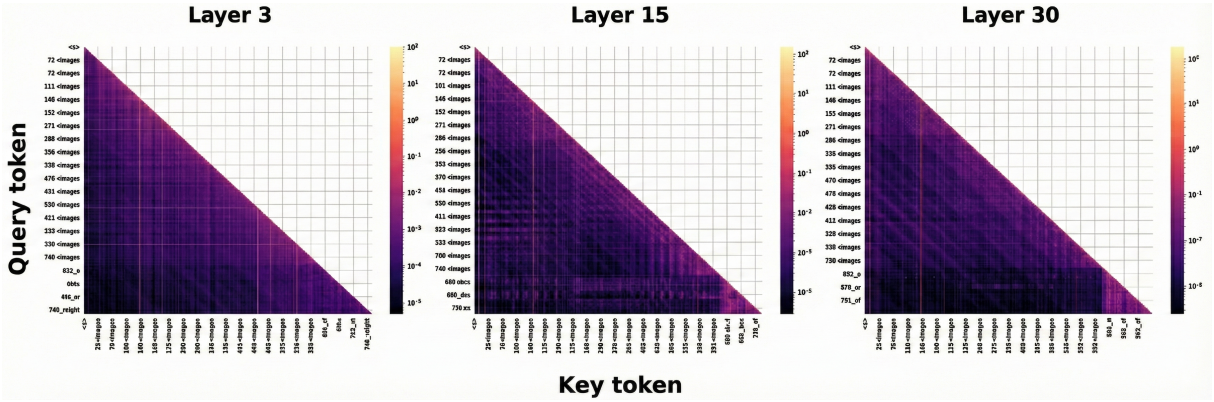

Figure 9: The layer-wise evolution of average attention contribution originating from query vectors and directed towards key vectors for all the tokens generated from a prompt "Describe the image in detail." across all layers of the in LLaVA-1.5.**Note: The contribution magnitude is plotted on a logarithmic scale**

by finding a learned scaling vector $\alpha \in \mathbb{R}^d$ that minimizes the reconstruction error between the approximated output and the true residual output of the FFN block.

Problem Formulation Let $x \in \mathbb{R}^d$ denote the input hidden state to a specific layer, and let $y \in \mathbb{R}^d$ denote the target output of the residual block, defined as $y = x + \text{FFN}(x)$. We seek an optimal scaling vector α such that the element-wise product $\hat{y} = x \odot \alpha$ approximates y with minimal error. We formulate this as an Ordinary Least Squares (OLS) regression problem. Since the Hadamard product operates independently on each feature dimension, we can decompose the optimization into d independent scalar problems. For a specific feature dimension $k \in \{1, \ldots, d\}$, we aim to minimize the sum of squared errors over a calibration dataset of N samples. The objective function $J(\alpha_k)$ for the k-th dimension is defined as:

$$J(\alpha_k) = \sum_{n=1}^{N} (\alpha_k x_{n,k} - y_{n,k})^2 \tag{14}$$

where $x_{n,k}$ and $y_{n,k}$ represent the values of the k-th feature for the n-th sample in the calibration set.

Optimization via Gradient Analysis To find the global minimum for this convex objective function, we compute the partial derivative of $J(\alpha_k)$ with respect to the parameter α_k and set it to zero:

$$\frac{\partial J}{\partial \alpha_k} = \sum_{n=1}^{N} 2(\alpha_k x_{n,k} - y_{n,k}) \cdot x_{n,k} = 0 \tag{15}$$

We can distribute the summation and factor out the constants to isolate α_k:

$$\sum_{n=1}^{N} (\alpha_k x_{n,k}^2 - x_{n,k} y_{n,k}) = 0 \tag{16}$$

$$\alpha_k \sum_{n=1}^{N} x_{n,k}^2 - \sum_{n=1}^{N} x_{n,k} y_{n,k} = 0 \tag{17}$$

Rearranging the terms yields the closed-form solution for the optimal scalar α_k:

$$\alpha_k = \frac{\sum_{n=1}^{N} x_{n,k} y_{n,k}}{\sum_{n=1}^{N} x_{n,k}^2} \tag{18}$$

This result confirms that the optimal scaling factor is simply the ratio of the cross-correlation between the input and target to the auto-correlation of the input.

Vectorized Implementation For efficient computation on GPUs, we vectorize this operation across all d dimensions simultaneously. Let $X \in \mathbb{R}^{N \times d}$ and $Y \in \mathbb{R}^{N \times d}$ be the matrices containing the collected calibration statistics. The optimal vector α^* can be computed using element-wise operations:

$$\alpha^* = \frac{\sum_{n=1}^{N} (x_n \odot y_n)}{\sum_{n=1}^{N} (x_n \odot x_n)} \tag{19}$$

where $\odot$ denotes the element-wise product and the division is performed element-wise. This analytical approach avoids the need for iterative gradient descent optimization. It allows us to calibrate the approximation parameters efficiently using only a small set of 500 samples, as the feature statistics converge rapidly to the optimal solution.

A.3 Evolutionary Trajectory of Visual Information Flow

It is crucial to analyze how the model's reliance on visual information evolves as data propagates through the layers. Fig. 9 visualizes the average attention contribution originating from query vectors and directed towards key vectors for all the tokens generated from a prompt "Describe the image in detail." across all layers of the LLaVA-1.5 architecture. This quantitative analysis reveals a distinct, progressive shift in the model's internal processing mechanism, characterized by three phases:

1. The Multimodal Fusion Phase (Early Layers). In the initial third of the network, we observe sustained, high-magnitude attention contributions from text to image tokens. During this stage, the model is actively engaged in grounding linguistic inputs within the visual context. The text tokens heavily query the visual features to resolve ambiguities and build an integrated multimodal representation.

2. The Abstraction Transition (Middle Layers). The middle layers mark a critical phase transition. The figure illustrates a precipitous drop in the average contribution score from text to image tokens. This indicates that the primary task of multimodal fusion is nearing completion. The essential semantic content from the image has been abstracted and integrated into the evolving textual representation. As a result, the raw visual tokens become less critical for immediate processing, though they may still be retained as fallback context.

3. The Deep Reasoning Phase (Late Layers). In the final layers, the attention contribution to image tokens nears zero. At this depth, the model operates almost exclusively on the fused, high-level textual representation. The generative process is driven by linguistic reasoning and next-token prediction dynamics, bearing strong resemblance to a pure large language model. The visual tokens at this stage which have any relevant information are Structural Anchors. This empirical observation provides the foundational justification for why aggressive pruning strategies even heuristic ones achieve near-lossless performance in the deepest layers of Large Multimodal Models. An interesting thing to note here is that this phenomenon only occurs in vision tokens and not in text tokens highlighting the high entropy on vision tokens as compared to text tokens and the sharp decline in contribution in deeper layers empirically validates the hypothesis that the model's dependency on raw visual tokens diminishes significantly after initial multimodal fusion, rendering them redundant in the final stages of processing.

A.4 Theoretical Analysis of Visual FFN Redundancy

In this section, we provide a theoretical framework to explain the layer-wise FFN redundancy profile observed in our empirical analysis. Specifically, we address why visual tokens consistently exhibit high cosine similarity (linearity) in the intermediate layers of the network, contrasting sharply with the non-linear evolution of text tokens. We ground this behavior in the Information Saturation Hypothesis and the functional staging of Multimodal Large Language Models.

A.4.1 The Read-Only Manifold Hypothesis

We posit that following the initial multimodal fusion phase, visual tokens and text tokens occupy functionally distinct manifolds within the residual stream. The text generation process represents a dynamic system where the text state must evolve layer-by-layer to reduce entropy for next-token prediction. This necessitates significant non-linear FFN transformations to disentangle semantic features and perform reasoning. In contrast, visual tokens in the intermediate layers serve primarily as conditioning constants or a static context for the text tokens. Once the visual features are projected into the semantic space of the language model during the initial layers, they must remain representationally stable. This stability is required to serve as a reliable addressable memory for the attention mechanism. If visual tokens were to undergo severe non-linear transformations via FFNs in every layer, the semantic alignment established in the early layers would drift, degrading the addressing system used by the text tokens to retrieve visual information. To maintain this stability, visual tokens in the middle layers converge to fixed points of the layer function. For a visual token x, the update rule approximates an identity mapping where the FFN output approaches zero or acts linearly. This constraint explains the high cosine similarity observed in the middle layers, as the residual connection dominates the update.

A.5 Functional Staging and Information Saturation

The observed redundancy profile aligns with the three-stage processing hierarchy identified in recent probing studies (Yu and Lee, 2025). We map the FFN redundancy dynamics to these functional stages:

Visual Grounding (Early Layers) In the initial layers, the model aligns visual features with the embedding space of the language model. While some non-linearity is required here to project features into the correct subspace, the redundancy metric rises quickly. This suggests that the projection head and the first few transformer layers handle the bulk of this alignment, rapidly stabilizing the visual representation.

Context Absorption and Saturation (Middle Layers) During the intermediate layers, the text tokens actively query the visual tokens to resolve semantic references and integrate multimodal information. We hypothesize that this phase is characterized by Information Saturation. Once the text tokens have absorbed the necessary visual context via cross-modal attention, the visual tokens effectively become informationally saturated sources. They are no longer the target of processing but rather the static reference. Mathematically, as the gradient of information flow becomes unidirectional (from image to text), the utility of FFNs for updating visual tokens diminishes. The FFNs thus collapse to linearity, validating why approximating these layers results in minimal performance loss.

Decoupling and Decoding (Late Layers) We observe a decline in cosine similarity for visual tokens in the final layers. This corresponds to the decoding phase described by (Yu and Lee, 2025), where the model shifts entirely to next-token prediction dynamics. At this stage, the attention mechanism often suppresses visual tokens as the model focuses on linguistic formatting and output generation. The divergence from linearity in these final layers likely reflects a semantic decoupling, where visual tokens are either transformed to disentangle task-specific features or simply drift due to a lack of attention constraints.