RedundancyLens: Revealing and Exploiting Visual Token Processing Redundancy for Efficient Decoder-Only MLLMs

Hongliang Li[1], **Jiaxin Zhang**[1], **Wenhui Liao**[1],
Dezhi Peng[1], **Kai Ding**[2,3], **Lianwen Jin**[*1,3]
[1]South China University of Technology [2]Intsig Information Co., Ltd.
[3]INTSIG-SCUT Joint Lab on Document Analysis and Recognition
eehongliangli@mail.scut.edu.cn, eejxzhang@gmail.com, eelwh@mail.scut.edu.cn
pengdzscut@foxmail.com, danny_ding@intsig.net, eelwjin@scut.edu.cn

Abstract

Current Multimodal Large Language Model (MLLM) architectures face a critical tradeoff between performance and efficiency: decoder-only architectures achieve higher performance but lower efficiency, while cross-attention-based architectures offer greater efficiency but lower performance. The key distinction lies in how visual tokens are processed. Decoder-only architectures apply self-attention and FFN operations on visual tokens, while cross-attention architectures skip these computations. To investigate whether redundancy exists in this computationally expensive process, we propose a training-free framework for analyzing trained MLLMs. It consists of Probe-Activated Dynamic FFN and Hollow Attention, which enable adjustable reductions in computations for visual tokens, as well as a Layer Ranking Algorithm that prioritizes layers for these reductions. Extensive experiments demonstrate substantial, structured, and clustered redundancy unique to decoder-only MLLMs, offering valuable insights for future MLLM architecture design. Furthermore, by leveraging our reduction framework as a training-free inference acceleration approach, we achieve performance comparable to or better than state-of-the-art methods while remaining compatible with them. Code is available at https://github.com/L-Hugh/RedundancyLens.

1 Introduction

Large Language Models (LLMs) (Brown et al., 2020; Zhang et al., 2022; Touvron et al., 2023; OpenAI, 2024) have seen rapid advancement in recent years, attracting attention for their strong capabilities in language comprehension and reasoning. In computer vision, researchers extend LLMs with visual abilities aimed at developing Multimodal Large Language Models (MLLMs) (Li et al., 2023; Alayrac et al., 2022; Liu et al., 2023; Zhu et al., 2024). These models hold significant potential for multimodal task solving and have become a prominent focus of current research. A key challenge in this area is designing effective architectures to integrate visual signals into LLMs.

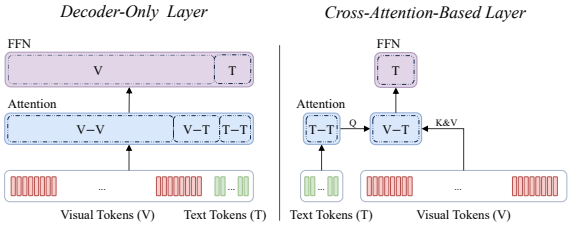


Figure 1: Comparison between decoder-only and cross-attention-based architectures from a unified perspective. Self-attention and FFN operations for visual tokens dominate the computation of decoder-only layers.

Current MLLMs are commonly built using either decoder-only (e.g., LLaVA (Liu et al., 2023)) or cross-attention-based architectures (e.g., Flamingo (Alayrac et al., 2022)). In the early development of MLLMs, the simplicity and effectiveness of the decoder-only architecture leads to its widespread adoption (Lu et al., 2024; Guo et al., 2025; Wei et al., 2025; Chen et al., 2024d). To capture finer-grained visual details, decoder-only MLLMs progressively increase input image resolutions, resulting in significant performance gains (Li et al., 2024b; Ye et al., 2023; Dong et al., 2024). However, this also leads to longer visual token sequences, significantly degrading the model's efficiency. Consequently, cross-attention-based architectures are attracting increasing interest due to their greater efficiency in handling long visual token sequences (Llama-Team, 2024; Chen et al., 2024a). Nonetheless, recent work (Dai et al., 2024) demonstrates that, decoder-only MLLMs tend to achieve significantly better overall performance. Developing an MLLM architecture that achieves both high performance and efficiency remains an

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 24056–24067

important area for further research.

To advance this research, this paper investigates MLLM architectures by evaluating existing designs and analyzing their redundancy patterns. We begin by comparing these two common architectures from a unified perspective. As shown in Figure 1, the fundamental difference between them lies in the processing of visual tokens: in decoder-only architectures, visual tokens undergo self-attention and FFN operations, whereas cross-attention-based architectures omit these operations. Since visual tokens often outnumber text tokens significantly (Dong et al., 2024), the processing of visual tokens consumes the majority of computational resources (roughly estimated as the ratio of visual tokens to total tokens, typically exceeding 90%). Investigating whether redundancy exists in this computationally expensive process is valuable. Considering the computational cost can be expressed as the number of layers multiplied by the cost of performing self-attention and FFN operations on visual tokens at each layer, the question arises: *are full self-attention and FFN operations for visual tokens required at every layer?*

Given the significant training costs of state-of-the-art MLLMs, we propose a training-free framework to investigate this question by analyzing trained decoder-only MLLMs. Specifically, we apply computational reductions to visual token processing in a subset of layers and evaluate their impact on model performance. By gradually increasing the number of layers where these reductions are applied, from a single layer to all layers, we can obtain a performance variation curve that reflects the degree of redundancy in the self-attention and FFN operations across layers.

To achieve this, the proposed framework consists of two components: (1) Probe-Activated Dynamic FFN and Hollow Attention, which replace the original FFN and attention modules, enabling adjustable reductions in computations for visual tokens. Specifically, Probe-Activated Dynamic FFN dynamically selects a subset of FFN parameters to process visual tokens. The Probe-Activated strategy is proposed to enable this selection in a training-free manner. Hollow Attention limits global attention among visual tokens to local attention while preserving attention between visual and text tokens. (2) Layer Ranking Algorithm, which assigns a rank to each layer. When selecting a subset of layers for computational reductions during the traversal process, those with the highest ranks are prioritized.

We conduct extensive experiments on state-of-the-art MLLMs, including InternVL2-8B (Chen et al., 2024c), Qwen2-VL-7B (Wang et al., 2024a), MiniCPM-V 2.6 (Yao et al., 2024), and LLaVA-OneVision (Li et al., 2024a). Our experiments are divided into two parts.

For the first part, the results show that applying the proposed reductions to approximately half of the layers preserves or even improves model performance. Notably, further applying these reductions to text tokens leads to a sharp decline in model performance. These findings reveal that decoder-only MLLMs exhibit substantial redundancy in the processing of visual tokens within certain layers. This structured and clustered redundancy can be effectively leveraged, providing valuable insights for future architecture design.

For the second part, leveraging our reduction framework as a training-free inference acceleration approach, we achieve performance comparable to or better than current state-of-the-art methods (Chen et al., 2025; Lin et al., 2024). Furthermore, existing approaches accelerate models by reducing the number of visual tokens, while our approach reduces the computational cost per visual token. Since these two methods are orthogonal, they can be combined for further acceleration.

In conclusion, our contributions are three-fold:

- We propose a framework to investigate redundancy in visual token processing through the analysis of trained decoder-only MLLMs.
- We demonstrate substantial, structured, and clustered redundancy unique to decoder-only MLLMs, offering valuable insights for future MLLM architecture design.
- We introduce a training-free MLLM acceleration method that takes a distinct and orthogonal perspective from current state-of-the-art methods, achieving comparable or better results while remaining compatible with them.

2 Related Work

2.1 MLLM Architectures

The decoder-only architecture is one of the most widely adopted designs for MLLMs (Li et al., 2023; Liu et al., 2023; Zhu et al., 2024), favored for its simplicity and efficiency. In this architecture, image tokens are concatenated with text token sequences and processed uniformly alongside text tokens by the LLM. A projector module maps the features extracted by the image encoder into the

input image tokens for the LLM, implemented using either a multilayer perceptron (Liu et al., 2023; Wang et al., 2024b; Lu et al., 2024) or cross-attention mechanisms (Li et al., 2023; Bai et al., 2023; Ye et al., 2024b). To improve fine-grained visual perception by capturing more detailed visual features, models like UReader (Ye et al., 2023) and Monkey (Li et al., 2024b) divide high-resolution images into multiple sub-images and concatenate their tokens for input into the LLM. Extending this idea, InternLM-XComposer2-4KHD (Dong et al., 2024) enhances the model's resolution capabilities to 4K HD and beyond, demonstrating consistent performance improvements. These advances have significantly accelerated the development of MLLMs (Chen et al., 2024c; Yao et al., 2024; Hong et al., 2024; Wang et al., 2024a; Li et al., 2024a), allowing open-source models to match or even surpass commercial multimodal models. However, increasing image resolution and multi-image input scenarios lead to longer input sequences, which significantly increase inference times and limit practical applications.

The cross-attention-based architecture offers greater efficiency in handling long visual token sequences, gaining increasing attention as an alternative to the decoder-only architecture. These architectures introduce additional cross-attention layers within the LLM to integrate visual information by applying cross-attention to visual tokens, thereby eliminating the need for the entire LLM to process them. Flamingo (Alayrac et al., 2022) is a prominent early work in this area, using a perceiver resampler to downsample the vision encoder's features before feeding them into the LLM via gated cross-attention layers. Llama 3-V (Llama-Team, 2024) adopts a similar structure but removes the perceiver module. EVLM (Chen et al., 2024a) utilizes hierarchical ViT features and a mixture of experts to enhance performance. mPLUG-Owl3 (Ye et al., 2024a) incorporates cross-attention mechanisms in parallel with self-attention layers instead of adding additional cross-attention layers. EE-MLLM (Ma et al., 2024) modifies the original self-attention mechanism into a composite attention mechanism. Meanwhile, NVLM (Dai et al., 2024) introduces a hybrid architecture that uses the LLM's self-attention layers to process thumbnail image tokens while employing cross-attention to capture finer image details.

To provide a fair comparison of the two architectures, recent work (Dai et al., 2024) trained both a decoder-only MLLM (NVLM-D) and a cross-attention-based MLLM (NVLM-X) under the same conditions. The results show that NVLM-X provides superior computational efficiency for high-resolution images, whereas NVLM-D delivers better overall performance. This comparison provides valuable insights for future research; however, further investigation at a more granular level would be beneficial.

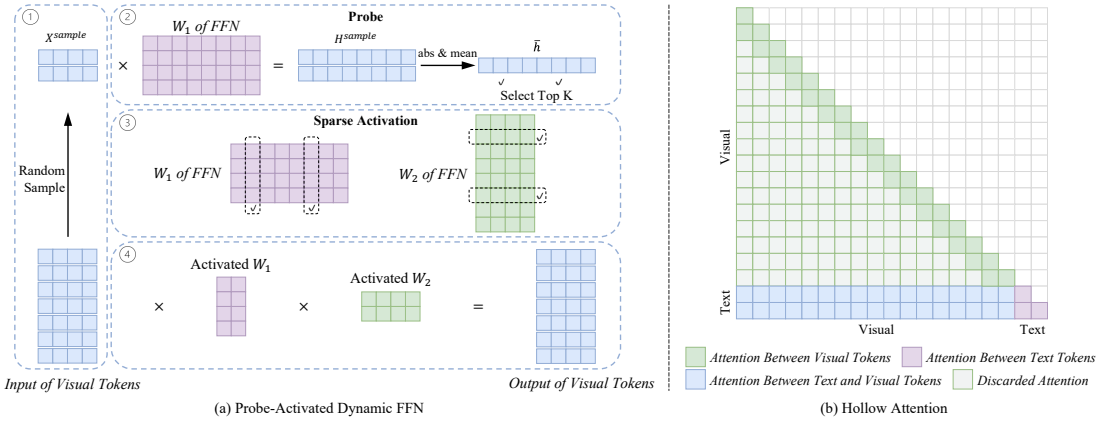
2.2 Visual Token Compression in MLLMs

Compressing visual sequence length is an effective and common method for accelerating MLLMs (Liu et al., 2024a; Zhang et al., 2024; Xing et al., 2024; Huang et al., 2024; He et al., 2024). Common techniques include using a group of learnable query tokens to extract information via cross-attention (Dai et al., 2023; Li et al., 2023; Alayrac et al., 2022), directly concatenating adjacent tokens (Chen et al., 2023; Yu et al., 2024), or downsampling through convolutional neural networks (Cha et al., 2024; Hu et al., 2024). Some recent approaches dynamically discard nonessential tokens during inference (Shang et al., 2024; Chen et al., 2025; Lin et al., 2024). For instance, FastV (Chen et al., 2025) reduces computational costs dramatically by pruning visual tokens based on their average attention scores at a selected layer in the MLLM, without sacrificing performance.

Token compression methods accelerate MLLMs by reducing the number of visual tokens, but the remaining tokens still require substantial computation in the LLM module, similar to text tokens. In contrast, our method achieves acceleration by reducing computation per visual token. This means our method is orthogonal to these methods and can be combined with them for further acceleration.

3 Methodology

The proposed framework consists of two components: (1) Probe-Activated Dynamic FFN and Hollow Attention, which replace the original FFN and attention modules, enabling adjustable reductions in computations for visual tokens. (2) Layer Ranking Algorithm, which assigns a rank to each layer. When selecting a subset of layers for computational reductions, those with the highest ranks are prioritized.



(a) Probe-Activated Dynamic FFN (b) Hollow Attention

Figure 2: Illustration of the proposed computational reductions for visual tokens: (a) Probe-Activated Dynamic FFN and (b) Hollow Attention. The symbol '×' denotes matrix multiplication.

3.1 Computational Reductions for Visual Tokens

3.1.1 Probe-Activated Dynamic FFN

Inspired by MoE (Shazeer et al., 2016), we reduce FFN computations for visual tokens by structurally activating only a subset of FFN parameters. However, we cannot directly adopt MoE, as it requires training a router that dynamically selects which parameters to activate. To achieve this without additional training, we propose Probe-Activated Dynamic FFN.

For each forward pass, the visual input $X \in \mathbb{R}^{N \times d_{\text{model}}}$ consists of N visual tokens, derived from a single image or multiple cropped images, where d_{model} is the feature dimension. The vanilla FFN layer (Vaswani et al., 2017) performs the following operations to obtain the output Y:

$$H = \text{ReLU}(XW_1 + \mathbf{b_1}) \in \mathbb{R}^{N \times d_{\text{ff}}}, \quad (1)$$

$$Y = HW_2 + \mathbf{b_2} \in \mathbb{R}^{N \times d_{\text{model}}}, \quad (2)$$

where $W_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$ and $W_2 \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$ are the weight matrices.

In the proposed Probe-Activated Dynamic FFN, we first randomly sample a subset $X^{\text{sample}} \in \mathbb{R}^{M \times d_{\text{model}}}$ from X, where M ($M \ll N$) denotes the number of sampled tokens. This sampled subset is used to compute the hidden representation:

$$H^{\text{sample}} = \text{ReLU}(X^{\text{sample}} W_1 + \mathbf{b_1}) \in \mathbb{R}^{M \times d_{\text{ff}}}. \quad (3)$$

We then take the element-wise absolute value of each token's hidden representation and compute the mean across the sampled tokens:

$$\bar{\mathbf{h}} = \frac{1}{M} \sum_{i=1}^{M} \left| H_i^{\text{sample}} \right| \in \mathbb{R}^{d_{\text{ff}}}. \quad (4)$$

Next, we select the top K elements from $\bar{\mathbf{h}}$ with the highest values. Let S represent the set of selected indices:

$$S = \text{Top}_K(\bar{\mathbf{h}}). \quad (5)$$

Using the selected indices S, we activate a subset of the weight matrices W_1 and W_2 as follows:

$$W_1^{\text{act}} = W_1[:, S] \in \mathbb{R}^{d_{\text{model}} \times K}, \quad (6)$$

$$W_2^{\text{act}} = W_2[S, :] \in \mathbb{R}^{K \times d_{\text{model}}}. \quad (7)$$

The corresponding bias $\mathbf{b_1}^{\text{act}}$ is activated similarly:

$$\mathbf{b_1}^{\text{act}} = \mathbf{b_1}[S] \in \mathbb{R}^{K}. \quad (8)$$

Finally, the forward propagation proceeds as follows:

$$H^{\text{act}} = \text{ReLU}(XW_1^{\text{act}} + \mathbf{b_1}^{\text{act}}) \in \mathbb{R}^{N \times K}, \quad (9)$$

$$Y = H^{\text{act}} W_2^{\text{act}} + \mathbf{b_2} \in \mathbb{R}^{N \times d_{\text{model}}}. \quad (10)$$

Figure 2 (a) provides a more intuitive illustration of the computation process in Probe-Activated Dynamic FFN, with activation functions and biases omitted for simplicity. It is important to note that this process applies only to visual tokens, while the FFN for text tokens remains unchanged. Some MLLMs modify the vanilla FFN, such as by adding gating mechanisms (Wang et al., 2024a; Chen et al., 2024c), yet our method can still be directly applied in these cases.

3.1.2 Hollow Attention

Inspired by sparse attention (Zaheer et al., 2020), we introduce a custom sparse attention pattern for MLLMs, called Hollow Attention, to reduce the

Algorithm 1 Layer Ranking Search

Input: Number of layers L, validation set
Output: Ranked list of layer indices
$RankedLayers \leftarrow []$
$UnrankedLayers \leftarrow \{1, 2, \ldots, L\}$
while $UnrankedLayers \neq \emptyset$ **do**
    $SelectedLayer \leftarrow null$
    $BestPerformance \leftarrow -\infty$
    for each $layer$ in $UnrankedLayers$ **do**
        Apply reduction to $RankedLayers \cup \{layer\}$
        Evaluate the model on the validation set
        and store the performance metric as P
        if $P > BestPerformance$ **then**
            $BestPerformance \leftarrow P$
            $SelectedLayer \leftarrow layer$
        end if
    end for
    $RankedLayers.append(SelectedLayer)$
    $UnrankedLayers.remove(SelectedLayer)$
end while
Return: $RankedLayers$

attention computation for visual tokens. As illustrated in Figure 2 (b), global attention among visual tokens is replaced with local attention, while the attention between visual and text tokens, as well as within text tokens, remains unchanged. Specifically, each visual token attends to the preceding R_A visual tokens (where R_A denotes the attention range) and all text tokens, whereas text tokens retain the ability to attend to all tokens. Since visual tokens typically outnumber text tokens by a large margin in MLLMs, this reduction effectively eliminates the majority of the attention overhead.

3.2 Layer Ranking Algorithm

Given the number of layers requiring reduction, denoted by L_r (where $0 \leq L_r \leq L$, and L is the total number of layers), the goal is to select the L_r-layer combination with the highest redundancy. To this end, we construct a compact validation set and use the performance variations of the MLLM on it to estimate redundancy. Since exhaustively evaluating all possible layer combinations for each value of L_r is computationally infeasible, we propose a search algorithm that ranks each layer, as detailed in Algorithm 1. For a given L_r, the top-ranked L_r layers are selected for reduction.

The validation set used in the Layer Ranking Algorithm comprises multiple subsets drawn from different datasets. The overall evaluation metric is computed by summing the scores across all subsets. For each subset, we compute the difference by subtracting the evaluation metric of the original model from that of the reduced model. If this difference is negative—indicating that the reduced model performs worse—it is multiplied by a penalty coefficient $\alpha > 1$. This penalization mechanism encourages the search process to prioritize reductions that maintain performance stability.

For each MLLM, Algorithm 1 is applied separately for FFN and attention reductions, with the process run twice. In our experiments, we observe that the last few layers of MLLMs tend to exhibit greater redundancy, making them a priority for reduction. To reduce the number of evaluations, we limit the ranking algorithm's search to the first $L - L_p$ layers. The last L_p layers are ranked in descending order of their position, starting from the last layer.

4 Experiments

4.1 Datasets

To construct the validation set for the Layer Ranking Algorithm, we randomly sample 750 instances from the full evaluation dataset collected in (Liu et al., 2024c), 1,000 instances from MMBench-DEV-EN-V11 (Liu et al., 2024b), and 200 instances each from the validation sets of DocVQA (Mathew et al., 2021), InfoVQA (Mathew et al., 2022), and ChartQA (Masry et al., 2022). To avoid overlap with the test set, when sampling the 750 instances from the full evaluation dataset in (Liu et al., 2024c), we exclude any samples that appear in OCR-Bench (Liu et al., 2024c), as well as those from TextVQA (Singh et al., 2019), DocVQA (Mathew et al., 2021), InfoVQA (Mathew et al., 2022), and ChartQA (Masry et al., 2022).

For evaluation, we conduct experiments on eight widely used benchmarks: OCRBench (Liu et al., 2024c), DocVQA (Mathew et al., 2021), InfoVQA (Mathew et al., 2022), ChartQA (Masry et al., 2022), TextVQA (Singh et al., 2019), MME (Fu et al., 2024), MMStar (Chen et al., 2024b), and HallusionBench (Guan et al., 2024). In particular, following prior work (Chen et al., 2024c; Wang et al., 2024a), we use the validation set of TextVQA (Singh et al., 2019) for evaluation.

4.2 Evaluation Metrics for Each Benchmark

We assess model performance using the standard metrics provided by each benchmark. OCR-Bench (Liu et al., 2024c) uses the number of correctly generated answers as its evaluation metric. DocVQA (Mathew et al., 2021) and InfoVQA (Mathew et al., 2022) use Aver-

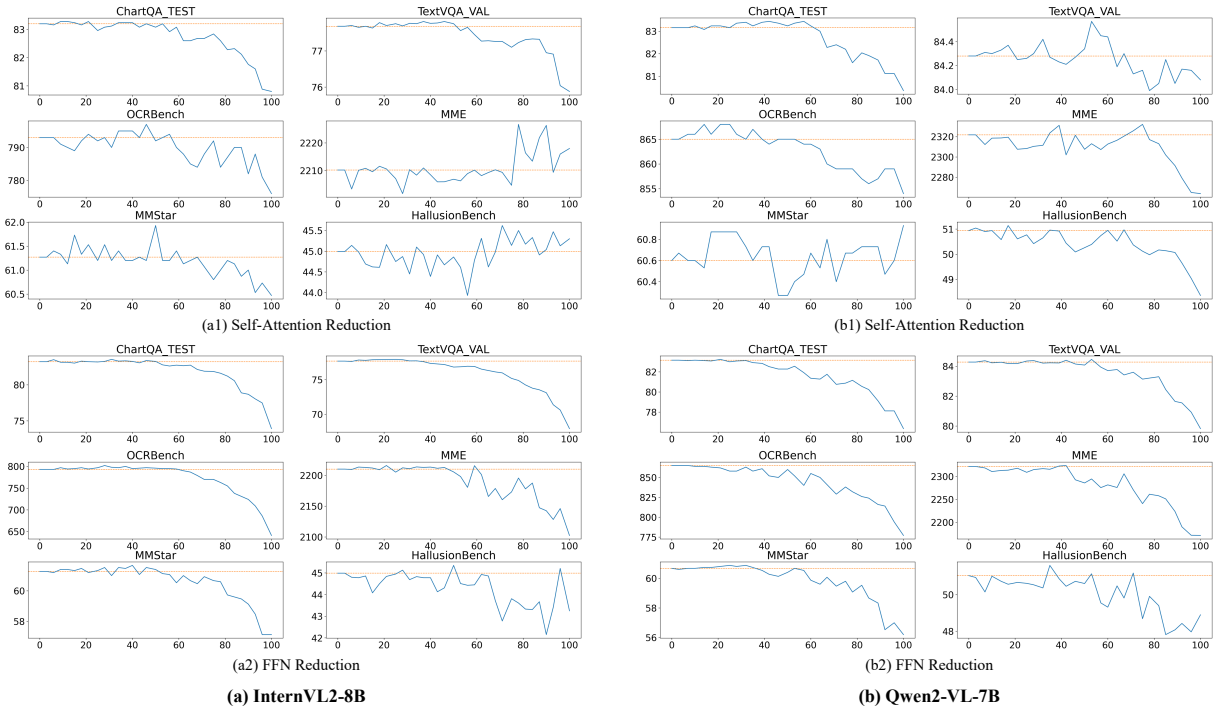


Figure 3: Impact of applying self-attention or FFN reductions across various layer proportions. The x-axis represents the percentage of layers with reductions applied, and the y-axis indicates model performance on the benchmark metric. The horizontal line shows the model's original performance (y-value at x=0).

age Normalized Levenshtein Similarity (ANLS) and are evaluated on their respective official websites. ChartQA (Masry et al., 2022) measures performance with relaxed accuracy, while TextVQA (Singh et al., 2019) relies on VQA accuracy (Goyal et al., 2017). MME (Fu et al., 2024) reports the sum of perception and cognition scores. MMStar (Chen et al., 2024b) evaluates models based on overall accuracy. HallusionBench (Guan et al., 2024) reports the average of Question Pair Accuracy, Figure Accuracy, and Overall Accuracy. Evaluation for all benchmarks, except DocVQA and InfoVQA, is conducted using VLMEvalKit (Duan et al., 2024).

4.3 Implementation Details

All experiments are conducted on NVIDIA A100 GPUs using VLMEvalKit (Duan et al., 2024), a framework for evaluating MLLMs on diverse multimodal benchmarks. We evaluate state-of-the-art MLLMs, including InternVL2-8B (Chen et al., 2024c), Qwen2-VL-7B (Wang et al., 2024a), MiniCPM-V 2.6 (Yao et al., 2024), and LLaVA-OneVision (Li et al., 2024a). In Hollow Attention, the attention range R_A for visual tokens is set to 256, which typically corresponds to the number of tokens in a single sub-image. In Probe-Activated Dynamic FFN, the number of randomly sampled visual tokens M is set to 10% of the total visual tokens per sample, while the number of activated parameters K is set to 20% of the original parameter count. For the Layer Ranking Algorithm, the penalty coefficient α is set to 2. During the ranking process, M is set to 100% of the total visual tokens per sample to minimize fluctuations caused by uncertainty.

4.4 Redundancy Analysis in FFN and Attention for Visual Tokens

First, we independently analyze the redundancy of FFN and self-attention operations on visual tokens in existing decoder-only MLLMs. Specifically, we apply computational reductions for FFN or self-attention in a subset of layers and evaluate the MLLM's performance on multiple mainstream benchmarks. By gradually increasing the number of layers where these reductions are applied, from a single layer to all layers, we obtain performance variation curves that reflect the degree of redundancy in the self-attention and FFN operations across layers.

The experimental results are shown in Figure 3. The results indicate that applying the proposed reductions to about half of the layers maintains the

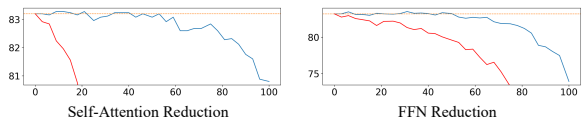


Figure 4: Performance comparison of the reductions applied to visual tokens (blue line) versus all tokens (red line), evaluated on ChartQA by InternVL2-8B.

MLLM's performance at a level comparable to the original model on most benchmarks, and in some cases, even surpasses the unreduced baseline. This outcome holds true for both the InternVL2 (Chen et al., 2024c) and Qwen2-VL (Wang et al., 2024a), regardless of whether the reductions are applied to self-attention or FFN operations. However, when reductions are applied to more than half of the layers, the performance of the MLLMs begins to decline rapidly across most benchmarks, with FFN reductions causing a sharper drop than self-attention reductions. In addition, further applying these reductions to text tokens leads to a sharp decline in model performance, as shown in Figure 4. These findings reveal that decoder-only MLLMs exhibit substantial redundancy in the processing of visual tokens within certain layers.

Current state-of-the-art MLLMs are built on pre-trained LLMs and fine-tuned on vast multimodal datasets, such as Qwen2-VL (Wang et al., 2024a), trained on over 1.4 trillion tokens of multimodal data. Therefore, the redundancy observed in processing visual tokens within LLMs cannot be attributed solely to insufficient training. We argue that this redundancy arises more from the inherent differences between visual and text tokens. On one hand, visual and text tokens originate from different modalities; on the other, visual tokens undergo extensive processing through an image encoder, while text tokens are processed only through linear mapping. These differences suggest that treating them equivalently within the LLM may not be the most efficient approach, especially considering the high computational demands of MLLMs in practice. By highlighting such redundancy, we hope to provide valuable insights for future architecture design.

4.5 Comparison with Training-free MLLM Inference Acceleration Methods

Building on the previous conclusions, our framework can accelerate decoder-only MLLM inference in a training-free manner, with the reduction pattern

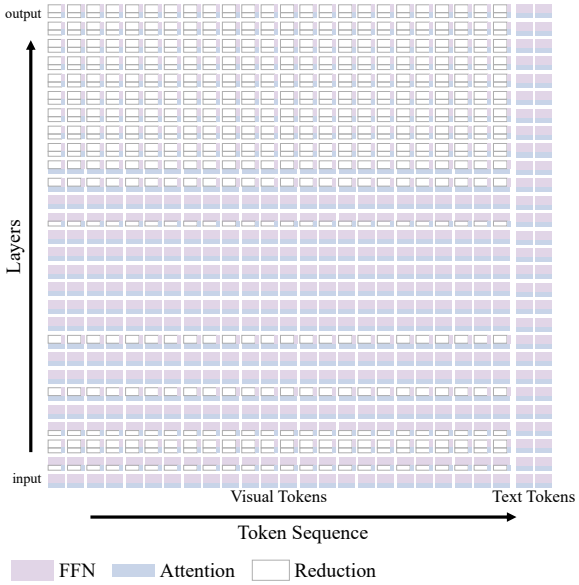


Figure 5: The reduction pattern for Qwen2-VL-7B. Notably, the attention from text tokens to all visual tokens needs to be preserved.

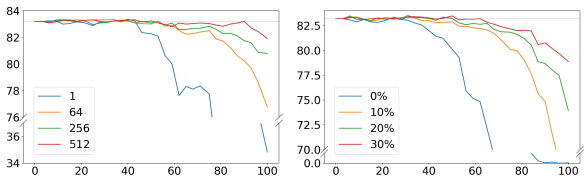
shown in Figure 5. We compare our approach with current state-of-the-art methods, which achieve acceleration by compressing the number of visual tokens, specifically FastV (Chen et al., 2025) and VTW (Lin et al., 2024). As shown in Table 1, our approach achieves comparable or superior performance to these token compression methods while reducing floating point operations (FLOPs) by approximately 30%. Additionally, our method and token compression methods address acceleration from different perspectives: token compression methods aim to reduce the number of visual tokens, whereas our approach focuses on lowering the computation required per visual token. This distinction indicates that the two methods are orthogonal and can be combined to achieve further acceleration. Table 1 demonstrates this synergy: when applying our approach alongside FastV (Chen et al., 2025) to reduce FLOPs by about 50%, model performance significantly surpasses that of FastV (Chen et al., 2025) alone with a higher compression rate across most benchmarks.

It is important to emphasize that our primary objective is to demonstrate the effectiveness of per-token computation reduction as an alternative acceleration approach, rather than to establish its superiority over token compression approaches. In fact, each approach is suited to different scenarios, as shown in Table 1. In cases of high information density within images, especially in text-rich contexts like OCRBench (Liu et al., 2024c), the potential for

Method	FLOPs Ratio	OCRBench	DocVQA	InfoVQA	ChartQA	TextVQA	MME	MMStar	HallusionBench
InternVL2-8B (32 Layers)	100%	793	91.6	74.4	83.2	77.7	2210	61.3	45.0
+ VTW (Lin et al., 2024) (L=23)	72%	704	87.9	69.3	80.0	69.3	2201	61.2	44.6
+ FastV (Chen et al., 2025) (K=2, R=30%)	72%	793	90.6	71.6	82.9	**77.6**	2181	60.7	45.3
+ **Ours** (L_{RA}=16, L_{RF}=17)	72%	**801**	**91.3**	**74.4**	**83.1**	77.2	**2212**	**61.7**	**45.6**
+ VTW (Lin et al., 2024) (L=17)	53%	64	14.5	30.9	17.9	20.8	**2200**	59.3	44.8
+ FastV (Chen et al., 2025) (K=2, R=50%)	53%	768	85.4	66.1	80.6	**77.1**	2195	59.3	44.9
+ **Ours** + FastV (K=2, R=30%)	52%	**797**	**90.3**	**71.6**	**83.0**	**77.1**	2192	**60.9**	**45.9**
Qwen2VL-7B (28 Layers)	100%	865	94.5	76.6	83.2	84.3	2322	60.7	51.0
+ VTW (Lin et al., 2024) (L=20)	71%	41	13.7	31.1	19.4	15.9	**2311**	**60.7**	50.3
+ FastV (Chen et al., 2025) (K=2, R=30%)	72%	829	94.4	75.1	82.6	84.0	2306	59.9	49.8
+ **Ours** (L_{RA}=13, L_{RF}=14)	71%	**859**	**94.5**	**75.7**	**83.0**	**84.6**	2309	60.5	**51.1**
+ VTW (Lin et al., 2024) (L=15)	54%	36	8.4	23.8	16.6	13.7	2174	53.7	39.7
+ FastV (Chen et al., 2025) (K=2, R=50%)	53%	766	93.4	71.0	79.4	83.6	2309	58.6	49.3
+ **Ours** + FastV (K=2, R=30%)	53%	**832**	**94.3**	**74.3**	**81.8**	**84.2**	**2310**	**59.7**	**51.1**
LLaVA-OneVision-7B (28 Layers)	100%	623	87.5	65.0	80.4	76.0	2002	61.7	39.3
+ VTW (Lin et al., 2024) (L=20)	71%	47	15.2	30.5	19.8	15.3	1991	**61.4**	40.2
+ FastV (Chen et al., 2025) (K=2, R=30%)	72%	590	85.4	61.4	77.0	**75.1**	2007	60.1	40.1
+ **Ours** (L_{RA}=13, L_{RF}=14)	71%	**635**	**85.6**	**63.2**	**80.2**	75.0	**2019**	61.0	**40.3**
+ VTW (Lin et al., 2024) (L=15)	54%	34	10.5	25.0	18.4	14.2	1897	52.5	29.6
+ FastV (Chen et al., 2025) (K=2, R=50%)	53%	506	78.9	53.8	68.0	72.5	1974	57.9	39.2
+ **Ours** + FastV (K=2, R=30%)	53%	**597**	**84.0**	**59.3**	**76.8**	**74.4**	**2019**	**59.9**	**40.2**
MiniCPM-V 2.6 (28 Layers)	100%	846	90.6	64.6	80.4	79.2	2276	57.5	48.3
+ VTW (Lin et al., 2024) (L=20)	71%	130	16.8	31.0	21.3	20.4	2250	57.3	34.9
+ FastV (Chen et al., 2025) (K=2, R=30%)	72%	800	85.5	59.3	78.4	79.0	2252	56.3	46.1
+ **Ours** (L_{RA}=13, L_{RF}=14)	71%	**847**	**90.0**	**63.8**	**79.8**	**79.6**	**2274**	**57.5**	**46.7**
+ VTW (Lin et al., 2024) (L=15)	54%	113	12.7	27.6	18.3	18.1	2053	53.5	30.1
+ FastV (Chen et al., 2025) (K=2, R=50%)	53%	749	72.5	52.2	72.9	77.0	2189	54.4	46.5
+ **Ours** + FastV (K=2, R=30%)	53%	**805**	**84.7**	**58.9**	**78.2**	**78.8**	**2228**	**55.1**	**46.7**

Table 1: Comparison of training-free methods for accelerating MLLM inference. The L_{RA} and L_{RF} in our method represent the number of layers for attention reduction and FFN reduction, respectively. FLOPs Ratio indicates the proportion of floating-point operations retained after applying the acceleration method compared to the full model. The best results are highlighted in bold, while the second-best results are underlined.

visual token compression is limited, making per-token computation reduction more effective. Conversely, in scenarios with lower information density, such as MME (Fu et al., 2024), reducing the number of visual tokens offers a higher upper bound for acceleration. This complementarity between approaches suggests that the optimal choice of acceleration strategy should be context-dependent, with the potential for combined implementation in hybrid solutions.



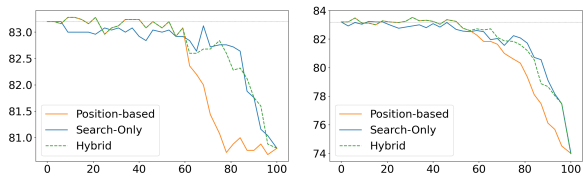
(a) Hollow Attention (b) Probe-Activated Dynamic FFN

Figure 6: Ablation studies on (a) the attention range in Hollow Attention and (b) the proportion of activated parameters in Probe-Activated Dynamic FFN, evaluated on ChartQA by InternVL2-8B. The x-axis indicates the percentage of layers with reductions applied, while the y-axis reflects the model's performance.

4.6 Ablation Studies

Ablation Studies on the Extent of Reduction in Self-Attention and FFN. To comprehensively evaluate the impact of reducing self-attention and FFN operations, we conduct ablation studies focusing on two factors: the proportion of activated parameters in Probe-Activated Dynamic FFN and the attention range in Hollow Attention, as illustrated in Figure 6. The results show that as the extent of reduction decreases (i.e., as the proportion of activated parameters or the attention range increases), more layers can be reduced without significantly impacting model performance. These two hyperparameters are selected by trading off efficiency and effectiveness.

Ablation Study of Layer Ranking Strategies. We compared three layer ranking strategies: (1) Position-based Strategy, which assigns the highest rank to the last layer and progressively decreases the rank toward the first layers; (2) Search-only strategy, which relies solely on Algorithm 1, with no layers pre-assigned ranks; and (3) Hybrid strategy, where the last L_p layers are pre-assigned the highest rank, with the remaining layers ranked by



(a) Self-Attention Reduction (b) FFN Reduction

Figure 7: Performance comparison of different layer ranking strategies, evaluated on ChartQA by InternVL2-8B. The first half of the orange line (Position-based) overlaps with and is obscured by the green line (Hybrid).

Algorithm 1. As shown in Figure 7, when reduction is applied to only a few layers, the position-based strategy outperforms the search-only strategy, indicating that later layers tend to exhibit higher redundancy in visual token processing. Additionally, the limited size of our validation set may not fully capture the true behavior of the models. As the number of reduced layers increases, the search-only strategy begins to yield better results. Therefore, we adopt the hybrid strategy, which combines the position-based strategy with the search-only strategy, to achieve better performance and reduce the number of evaluations required by Algorithm 1.

5 Conclusion

In this paper, we present a systematic investigation into the redundancy of visual token processing, which plays a crucial role in the trade-off between performance and efficiency in mainstream MLLM architectures. Through careful analysis of existing MLLMs, we propose a new framework consisting of two key components: computational reductions for visual tokens and a layer ranking algorithm. These reductions are applied across various layer proportions to evaluate their impact on MLLM performance. Extensive experiments reveal that current decoder-only MLLMs exhibit significant redundancy in visual token processing within certain layers. This structured and clustered redundancy can be effectively leveraged, providing valuable insights for future architectural design. Furthermore, this work opens new perspectives on training-free acceleration strategies for MLLMs, suggesting that future improvements in model efficiency might benefit from considering both token-level compression and computation-level optimization.

Limitations

Determining the layer rank for reduction through search in the validation set presents two limitations. First, it requires constructing a validation set and performing hundreds of evaluations. Additionally, to reduce computational resource demands, we use a limited-scale validation set and a greedy search-based algorithm, which may fail to identify the optimal combination of layers for reduction. Therefore, improvements to the Layer Ranking Algorithm or exploration of alternative features for determining layer reduction priorities warrant further investigation.

Acknowledgements

This research is supported in part by the National Natural Science Foundation of China (Grant No.: 62476093, 62441604).

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: A visual language model for few-shot learning. In *NeurIPS*, volume 35, pages 23716–23736.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*, volume 33, pages 1877–1901.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. 2024. Honeybee: Locality-enhanced projector for multimodal LLM. *arXiv preprint arXiv:2312.06742*.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. MiniGPT-v2: large language model

as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Kaibing Chen, Dong Shen, Hanwen Zhong, Huasong Zhong, Kui Xia, Di Xu, Wei Yuan, Yifei Hu, Bin Wen, Tianke Zhang, Changyi Liu, Dewen Fan, Huihui Xiao, Jiahong Wu, Fan Yang, Size Li, and Di Zhang. 2024a. EVLM: An efficient vision-language model for visual understanding. *arXiv preprint arXiv:2407.14177*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2025. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, pages 19–35.

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

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. 2024c. How far are we to GPT-4V? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024d. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, pages 24185–24198.

Wenliang Dai, Nayeon Lee, Boxin Wang, Zhuoling Yang, Zihan Liu, Jon Barker, Tuomas Rintamaki, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. NVLM: Open frontier-class multimodal LLMs. *arXiv preprint arXiv:2409.11402*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. InstructBLIP: towards general-purpose vision-language models with instruction tuning. In *NeurIPS*.

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Zhe Chen, Xinyue Zhang, Wei Li, Jingwen Li, Wenhai Wang, Kai Chen, Conghui He, Xingcheng Zhang, Jifeng Dai, Yu Qiao, Dahua Lin, and Jiaqi Wang. 2024. InternLM-XComposer2-4KHD: A pioneering large vision-language model handling resolutions from 336 pixels to 4K HD. *arXiv preprint arXiv:2404.06512*.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, Dahua Lin, and Kai Chen. 2024. VLMEvalKit: An open-source toolkit for evaluating large multi-modality models. In *ACM MM*, page 11198–11201.

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. *arXiv preprint arXiv:2306.13394*.

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 *CVPR*, pages 6904–6913.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. 2024. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *CVPR*, pages 14375–14385.

Zonghao Guo, Ruyi Xu, Yuan Yao, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. 2025. LLaVA-UHD: An LMM perceiving any aspect ratio and high-resolution images. In *ECCV*, pages 390–406.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. 2024. ZipVL: Efficient large vision-language models with dynamic token sparsification and KV cache compression. *arXiv preprint arXiv:2410.08584*.

Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. 2024. CogVLM2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. 2024. mPLUG-DocOwl 1.5: Unified structure learning for OCR-free document understanding. *arXiv preprint arXiv:2403.12895*.

Mingxin Huang, Yuliang Liu, Dingkang Liang, Lianwen Jin, and Xiang Bai. 2024. Mini-Monkey: Alleviating the semantic sawtooth effect for lightweight MLLMs via complementary image pyramid. *arXiv preprint arXiv:2408.02034*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, volume 202, pages 19730–19742.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2024b. Monkey: Image resolution and text label are important things for large multi-modal models. In *CVPR*, pages 26763–26773.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. 2024. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. *arXiv preprint arXiv:2405.05803*.

Haogeng Liu, Quanzeng You, Xiaotian Han, Yongfei Liu, Huaibo Huang, Ran He, and Hongxia Yang. 2024a. Visual anchors are strong information aggregators for multimodal large language model. *arXiv preprint arXiv:2405.17815*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *NeurIPS*, volume 36, pages 34892–34916.

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. 2024b. MMBench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Cheng lin Liu, Lianwen Jin, and Xiang Bai. 2024c. OCRBench: On the hidden mystery of OCR in large multimodal models. *arXiv preprint arXiv:2305.07895*.

Llama-Team. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. 2024. DeepSeek-VL: Towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*.

Feipeng Ma, Yizhou Zhou, Hebei Li, Zilong He, Siying Wu, Fengyun Rao, Yueyi Zhang, and Xiaoyan Sun. 2024. EE-MLLM: A data-efficient and compute-efficient multimodal large language model. *arXiv preprint arXiv:2408.11795*.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of ACL*, pages 2263–2279.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. 2022. InfographicVQA. In *WACV*, pages 2582–2591.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. DocVQA: A dataset for VQA on document images. In *WACV*, pages 2199–2208.

OpenAI. 2024. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2016. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*.

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 *CVPR*, pages 8309–8318.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, volume 30.

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

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2024b. CogVLM: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*.

Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, Jinrong Yang, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. 2025. Vary: Scaling up the vision vocabulary for large vision-language model. In *ECCV*, pages 408–424.

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

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. MiniCPM-V: A GPT-4V level MLLM on your phone. *arXiv preprint arXiv:2408.01800*.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, Qin Jin, Liang He, Xin Lin, and Fei Huang. 2023. UReader: Universal OCR-free visually-situated language understanding with multimodal large language model. In *Findings of EMNLP*, pages 2841–2858.

Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2024a. mPLUG-Owl3: Towards long image-sequence understanding in multi-modal large language models. *arXiv preprint arXiv:2408.04840*.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2024b. mPLUG-Owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Ya-Qi Yu, Minghui Liao, Jihao Wu, Yongxin Liao, Xiaoyu Zheng, and Wei Zeng. 2024. TextHawk: Exploring efficient fine-grained perception of multimodal large language models. *arXiv preprint arXiv:2404.09204*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for longer sequences. In *NeurIPS*, volume 33, pages 17283–17297.

Jiaxin Zhang, Wentao Yang, Songxuan Lai, Zecheng Xie, and Lianwen Jin. 2024. DocKylin: A large multimodal model for visual document understanding with efficient visual slimming. *arXiv preprint arXiv:2406.19101*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*.