arXiv:2410.06169v3 [cs.CV] 30 Nov 2024

Treat Visual Tokens as Text? But Your MLLM Only Needs Fewer Efforts to See

⋆Zeliang Zhang[1], ⋆Phu Pham[2] , ⋆Wentian Zhao[3]†, ⋆Kun Wan[3]†, Yu-Jhe Li[3], Jianing Zhou[4],
Daniel Miranda[3], Ajinkya Kale[3], Chenliang Xu[1]
[1]University of Rochester [2]Purdue University [3]Adobe Inc. [4] UIUC
{zeliang.zhang, chenliang.xu}@rochester.edu, phupham@purdue.edu
{wezhao, kuwan, jhel, miranda, akale}@adobe.com
zjn1746@illinois.edu

Abstract

By treating visual tokens from visual encoders as text tokens, Multimodal Large Language Models (MLLMs) have achieved remarkable progress across diverse visual understanding tasks, leveraging the robust architectures of Large Language Models (LLMs). However, as token counts grow, the quadratic scaling of computation in LLMs introduces a significant efficiency bottleneck, impeding further scalability. Although recent approaches have explored pruning visual tokens or employing lighter LLM architectures, the computational overhead from an increasing number of visual tokens remains a substantial challenge. It also raises challenges to decide important visual tokens for different sample, which introduces another computation overhead.

In this study, we investigate the redundancy in visual computation at both the parameter and computational pattern levels within LLaVA, a representative MLLM, and introduce a suite of streamlined strategies to enhance efficiency. These include neighbor-aware visual token attention, pruning of inactive visual attention heads, and selective layer dropping for visual computations. By implementing these strategies in LLaVA, we achieve a reduction in computational demands of 88% while maintaining model performance across key benchmarks. Additionally, we validate the existence of visual computational redundancy in other MLLMs, such as Qwen2-VL-7B and InternVL-2.0-4B/8B/26B. These results present a novel pathway for MLLMs to handle dense visual tokens with minimal computational costs. Our pruning method only requires one time to discover the important visual-related computation and is sample-agnostic. This indicates that, You Only need Pune Once (YOPO) to accelerate your MLLMs. Code and model checkpoints are available at `https://github.com/ZhangAIPI/YOPO_MLLM_Pruning`.

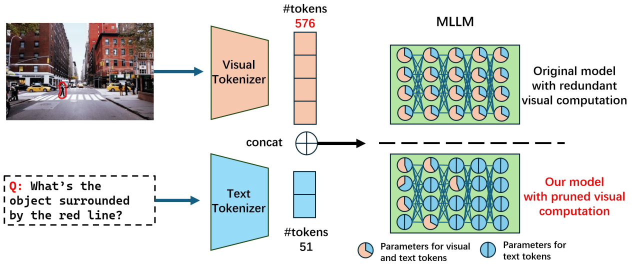

#tokens
576
Visual Tokenizer
MLLM
Original model with redundant visual computation
concat
Q: What's the object surrounded by the red line?
Text Tokenizer
#tokens
51
Our model with pruned visual computation
Parameters for visual and text tokens
Parameters for text tokens

Figure 1. Compared to text tokens processed by language prompts, visual tokens are significantly more numerous, leading to substantial computational overhead in MLLMs. However, visual information is often sparser, resulting in considerable redundancy within visual computations. In this work, we propose pruning these redundant computations at both the parameter and computational pattern levels to improve processing efficiency.

1. Introduction

After the tremendous success of LLMs [1, 26, 48, 51], researchers began exploring the introduction of additional modalities, such as images [2, 9, 28, 38], videos [33], and audio [13], into this architecture. Over the past two years, MLLMs, which incorporate more modalities, have been continuously improving performance. MLLMs based on the LLMs architecture have gained cross-modal understanding capabilities without significantly compromising the original abilities of LLMs. To further enhance the capabilities of MLLMs, various efforts are being made simultaneously, including architectural improvements [41, 49, 50, 54], data curation [5–7, 45], cross-modal alignment [2, 41, 54, 63], and training recipe optimizations [47, 60].

Recent works [27, 36, 53] demonstrate that as the scale of Large Language Models (LLMs) increases, so do the capabilities of Multimodal Large Language Models (MLLMs). Furthermore, other studies [14, 23] reveal that higher image resolution generally enhances performance. However, utilizing larger LLMs or incorporating additional visual tokens

---

⋆ indicates the equal contribution with random order.
† indicates the project leader.

imposes a significant computational burden. For instance, as shown in Fig. 1, a simple query in a visual question-answering task may involve only 51 text tokens but up to 576 visual tokens—over ten times the number of text tokens. This substantial imbalance, combined with the high volume of visual tokens, poses a critical challenge: the computational cost for LLMs rises quadratically with the total input tokens, thereby limiting the scalability of MLLMs.

To address these challenges, various approaches [11, 12, 35, 55, 59, 66, 67] have attempted to use more lightweight LLMs with fewer parameters, although these often experience noticeable performance drops on general tasks. Recognizing the substantial redundancy within visual representations, other methods [29, 32, 44, 58, 61, 62] aim to compress visual tokens to reduce input length for LLMs. However, determining which visual tokens are essential within the model remains an open question, and this compression frequently compromises model performance. Additionally, with the inclusion of video inputs, further reduction of image tokens becomes increasingly challenging.

In our work, we seek to fully leverage visual redundancy to enhance the efficiency of MLLMs. Rather than pruning input visual tokens to reduce redundant visual representations, we explore redundancy within visual computation patterns in MLLMs. Using LLaVA as a case study, we conduct a detailed analysis of redundancies in both visual attention and representation computations across all layers. Based on these insights, we introduce a set of simple yet effective strategies to significantly reduce the computational burden from visual information: neighbor-aware visual attention, non-active visual attention head dropping, sparse projection in the FFN, and lazy layer dropping for visual processing. As shown in Fig. 1, most parameters are dropped during visual computation while remaining active in text computation, thereby maintaining the strong understanding capability of the LLM while efficiently reducing the computational overhead introduced by the large number of visual tokens. By integrating these strategies into LLaVA, we reduce visual computational overhead by up to 88% while retaining nearly the same performance as the original model. Across multiple benchmarks, our method achieves state-of-the-art performance compared to various baselines. Furthermore, these findings are not unique to the LLaVA-based model; similar visual computational redundancies are also present in other MLLMs, including Qwen2-VL-7B and InternVL-2.0-4B/8B/26B. While token pruning-based methods require dynamic identification of important visual tokens on a per-request basis, our method directly prunes the computation pattern in MLLM from the outset of model development. This means that You Only Need to Prune Once (YOPO), which effectively reduces visual computation redundancy and accelerates MLLM performance across all subsequent use cases. These results highlight a promising direction for future research to enhance MLLM efficiency.

We summarize our contributions as follows:

1. We systematically study redundancy in the visual computation pattern of LLaVA, focusing on both intra-modal attention computation among visual tokens and cross-modal attention between visual and text tokens.
2. We introduce three strategies for visual computation pruning: neighbor-aware visual attention, non-active visual attention head dropping, sparse visual projection in the FFN, and lazy layer dropping for visual processing. These strategies effectively reduce computational costs at the inherent computation pattern level, alleviating the increasing visual computation overhead associated with a large number of visual tokens. Notably, our method prunes the MLLM only once, and it can be deployed for all subsequent computation requests.
3. In experiments, we apply these strategies to prune LLaVA, achieving state-of-the-art results on various key benchmarks with an 88% reduction in computational overhead across vision benchmarks while retaining most model performance. We further validate visual computation redundancy in other MLLMs, including Qwen2-VL-7B and InternVL-2.0-4B/8B/26B, showing that $25-50\%$ of visual computations from parameters and patterns can be pruned without fine-tuning while maintaining comparable performance.

2. Related work

2.1. Efficient MLLMs

Various approaches have been proposed to accelerate MLLMs, including designing smaller vision encoders [15, 17, 37], transforming the dense LLMs into the mixture-of-experts (MoE) [65], adopting compact language models [11, 22, 25], and implementing visual token pruning [31, 57, 61], among others. Among these, token pruning-based methods reduce the length of the input visual sequence without affecting model parameters or capacity, making them increasingly popular.

For instance, LLaVA-PruMerge [43] identifies and merges similar tokens at CLIP's penultimate layer, while PyramidDrop [56] applies a predefined token-dropping ratio across layers. Similarly, SparseVLM [64] progressively prunes visual tokens deemed irrelevant to corresponding text tokens, and FastV [8] enhances the computational efficiency of multi-modal LLMs by using adaptive attention patterns in initial layers to identify essential visual tokens, pruning less important ones in later layers. Although these approaches mitigate computational load, they often compromise model quality due to the loss of fine-grained visual details.

In our work, while we also leverage the sparsity of visual computation, we propose that pruning within the LLM itself offers a more efficient alternative. This approach preserves

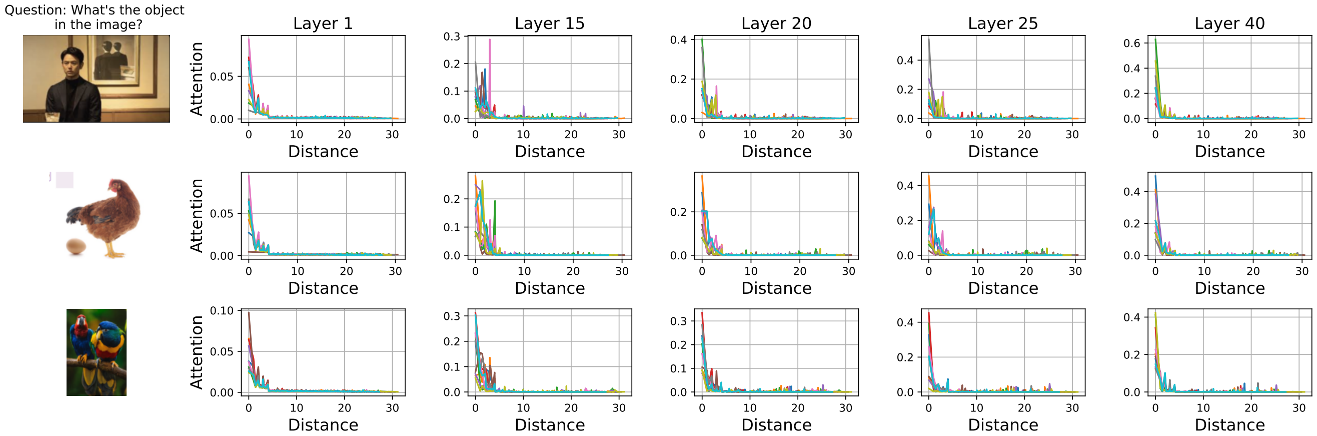


Figure 2. Visualization of attention weights for randomly selected vision tokens interacting with other visual tokens at varying spatial distances across different layers in LLaVA. Notably, the attention weights are predominantly concentrated on neighboring visual tokens.

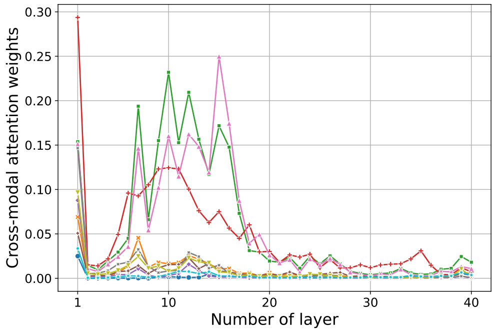


Figure 3. Visualization of the cross-modal attention weights between vision and text across varying layers. Each line represents the mean attention of an individual text token directed toward all other vision tokens, illustrating how attention varies as layers progress.

visual detail more effectively than methods that directly reduce the quantity of visual tokens.

2.2. LLM pruning

The quadratic complexity of LLMs [52] has made investigating and mitigating their computational redundancy a significant focus in the research community. Techniques such as pruning based on weight importance have been proposed by Han et al. [21] and Frantar and Alistarh [18], while Michel et al. [42] demonstrated that many attention heads can be removed to improve efficiency. Similarly, Fan et al. [16] introduced a method for randomly dropping transformer layers without impacting model performance. While computational redundancy in LLMs has been well-explored, adapting these methods to MLLMs remains an open challenge. In particular, optimizing LLM pruning techniques for the multimodal nature of MLLMs, especially with a focus on the vision modality, has yet to receive adequate attention.

Our work addresses this gap for the first time by proposing pruning optimizations specifically designed for the LLM components of MLLMs to enhance their efficiency in processing visual information.

3. Method

3.1. Redundancy of visual computation in MLLMs

In MLLMs, vision tokens are projected into the same latent space as text tokens, with both types processed equally within the network's computational flow. Specifically, given the vision token V_v with length N_v and text token V_t with length N_t, the major computation overhead in MLLMs falls on two modules, including the attention operation

$$\begin{aligned} &\text{Attention}(\mathbf{Q}_v, \mathbf{Q}_t, \mathbf{K}_v, \mathbf{K}_t, \mathbf{V}_v, \mathbf{V}_t) \\ &= \text{Softmax}\left(\frac{[\mathbf{Q}_v; \mathbf{Q}_t]\,[\mathbf{K}_v; \mathbf{K}_t]^T}{\sqrt{d}}\right)[\mathbf{V}_v; \mathbf{V}_t]\,, \end{aligned} \tag{1}$$

and the feed-forward layer,

$$\text{FFN}(H_v, H_t) = \sigma\left(\sigma\left([H_v; H_t]\, W_1\right) W_2\right), \tag{2}$$

where $\mathbf{Q}_v \in \mathbb{R}^{N_v \times d}$ and $\mathbf{Q}_t \in \mathbb{R}^{N_t \times d}$ are the queries for the vision and text tokens, $\mathbf{K}_v \in \mathbb{R}^{N_v \times d}$ and $\mathbf{K}_t \in \mathbb{R}^{N_t \times d}$ are the keys, $\mathbf{V}_v \in \mathbb{R}^{N_v \times d}$ and $\mathbf{V}_t \in \mathbb{R}^{N_t \times d}$ are the values, H_v and H_t are the visual and text hidden representation, $W_1 \in \mathbb{R}^{d \times d'}$ and $W_2 \in \mathbb{R}^{d' \times d}$ are the weight matrices for linear projection in FFN, $\sigma(\cdot)$ is the activation function. Given that the number of vision tokens often far exceeds that of text tokens, vision-related computations dominate the workload, causing computational costs to scale approximately quadratically with the number of vision tokens.

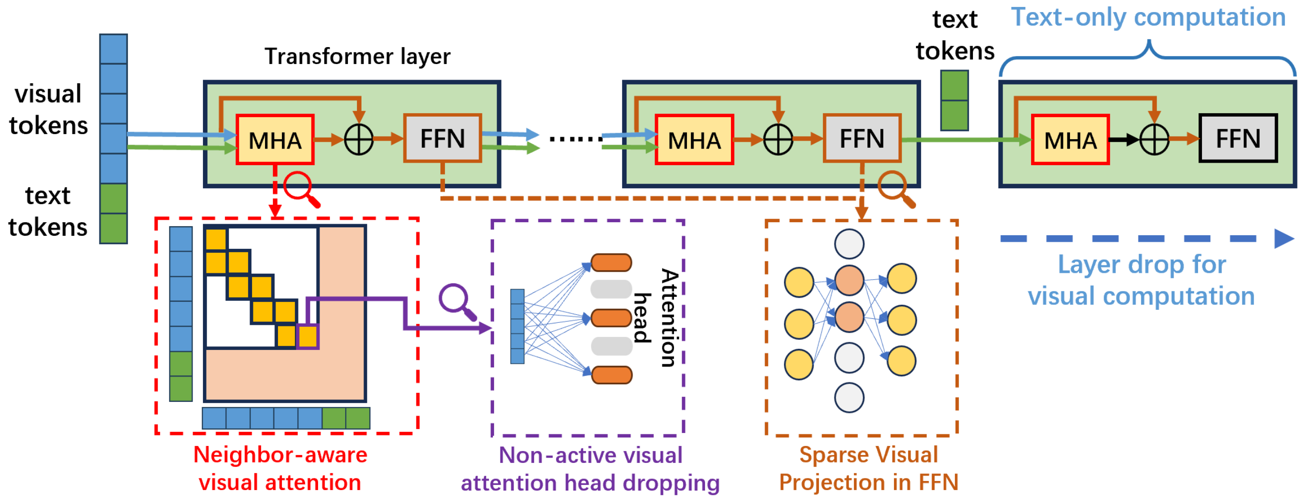

text tokens
Text-only computation
Transformer layer
visual tokens
text tokens
MHA
FFN
MHA
FFN
MHA
FFN
Attention head
Layer drop for visual computation
Neighbor-aware visual attention
Non-active visual attention head dropping
Sparse Visual Projection in FFN

Figure 4. Overview of our method. Our approach replaces the traditional attention block with a neighbor-aware visual attention mechanism, reducing computational complexity from quadratic to linear with respect to the number of visual tokens. Additionally, we prune inactive attention heads in the visual computation, focusing only on the most impactful components. To further decrease computation overhead, we disable visual processing in the later layers, where visual information has minimal impact on the task.

Despite this significant computational load, much of the visual processing in MLLMs is redundant due to the inherent spatial sparsity in visual data. To illustrate this, we randomly select 10 visual tokens from three example images and visualize their attention weights with respect to other vision tokens in shown images, *i.e.*, $Q_v K_v^T$ in Eq. (1). More specifically, we first compute the average attention weights across all attention heads and then rearrange these weights for each selected token based on its spatial distance from other tokens. This enables us to investigate the relationship between token spatial distance and the redundancy in visual attention computation. The spatial distance between visual token i and j is defined as follows,

$$\text{Distance}(i,j) = \sqrt{(i.x - j.x)^2 + (i.y - j.y)^2}, \quad (3)$$

where we denote x and y as the coordinates of the tokens in original 2D image.

The results in Fig. 2 show that high attention weights are predominantly concentrated among neighboring tokens, with values diminishing rapidly for tokens farther away. This pattern reflects significant sparsity, suggesting substantial potential for pruning.

This redundancy is observed not only within the vision modality but also in its interactions with text tokens, *i.e.*, $Q_t K_v^T$ in Eq. (1). As shown in Fig. 3, we further randomly select 10 vision tokens and compute the mean attention across other vision tokens at various layers. The results reveal high sparsity, with attention values generally decreasing to a low level after approximately the 20-th layer. These findings suggest that visual computation in MLLMs contains significant redundancy, presenting substantial opportunities for pruning to accelerate processing.

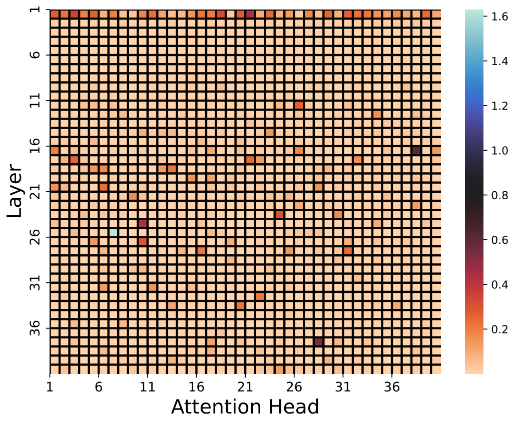


Figure 5. Visualization of ρ of different attention heads in different layers of LLaVA.

3.2. Neighbor tokens matter for visual attention

As analyzed in Fig. 2, although there are many visual tokens in MLLMs, most of the attention computation in $Q_v K_v^T$ is sparse, with significant attention weights concentrated primarily on neighboring visual tokens. To reduce the computational burden caused by this redundancy, we propose a simple yet effective pruning method that selectively eliminates non-essential attention computations among visual tokens. Specifically, we modify the attention mechanism so that only neighboring visual tokens attend to each other, while text tokens retain the ability to attend across both visual and text tokens. The modified attention computation for visual tokens can be formulated as follows:

$$\text{Attention}(\mathbf{Q}_v, \mathbf{K}_v, \mathbf{V}_v) = \text{Softmax}\left(\frac{\mathbf{Q}_v \mathbf{K}_v^T}{\sqrt{d}} + \mathbf{M}\right)\mathbf{V}_v,$$
$$s.t., \quad \mathbf{M}_{ij} = \begin{cases} 0 & \text{if } |i-j| \le h \\ -\infty & \text{if } |i-j| > h \end{cases}, \tag{4}$$

where h is the window size indicating the neighborhood of visual tokens. At the same time, the remaining attention computations for the visual-text and text-only tokens are computed as

$$\begin{aligned} &\text{Attention}(\mathbf{Q}_t, \mathbf{K}_v, \mathbf{K}_t, \mathbf{V}_v, \mathbf{V}_t) \\ &= \left[\text{Softmax}\left(\frac{\mathbf{Q}_t \left[\mathbf{K}_v; \mathbf{K}_t\right]^T}{\sqrt{d}}\right)[\mathbf{V}_v; \mathbf{V}_t]\right]. \end{aligned} \tag{5}$$

The results of Eq. (4) and Eq. (5) will be contacted as the output attention map with high sparsity.

The neighboring connections $\mathbf{M}$ in Eq. (4) can be identified through geometric computation within the CUDA kernel function during the runtime of matrix multiplication, allowing us to dynamically focus only on relevant tokens. This approach enables accelerated attention computation by limiting the number of involved tokens, thereby reducing the overall computational load. Specifically, when the number of visual tokens N_V is much larger than the number of text tokens N_T, the computational complexity reduces from $\mathcal{O}((N_V + N_T)^2)$ to $\mathcal{O}(N_V N_T + N_T^2)$, which grows linearly with N_V instead of quadratically.

Sparse Visual Projection in FFN By disabling most of the visual attention computation, the model's visual representation becomes highly sparse. To leverage this sparsity effectively, we also randomly drop $p\%$ of neurons in the hidden layer of the FFN within each block, *i.e.*, reducing the dimension d' used in W_1 and W_2 of Eq. (2).

3.3. Not all visual attention heads are equal

While we simplify visual attention by focusing only on neighboring visual tokens, the overall computation overhead for visual information remains substantial, largely due to multi-head attention. A natural question arises, *is every visual attention head equal in visual computation*?

Table 1. Evaluation results of dropping various ranges of transformer blocks for visual computation in LLaVA-1.5-13B. The model is not fine-tuned following visual computation pruning.

Range	GQA	SQA^I	TextVQA	MMB	Avg
Ori.	63.0	72.4	59.8	68.6	66.0
0–5	38.3	63.8	38.7	56.5	57.8
10–15	58.1	71.2	55.7	62.7	65.2
20–25	62.1	72.3	59.1	68.6	66.0
30–35	63.1	72.4	59.8	68.6	66.0
35–40	63.0	72.2	59.3	68.6	66.0

To address this question, we randomly select 100 samples from a visual-question answering task and forward them through LLaVA, recording the attention head values across different layers. To better quantify the impact of visual information on the task, we calculate the relative attention head value which is computed by the mean attention values for all visual tokens and all text tokens, respectively, which can be formulated as follows,

$$\rho^h = \frac{\frac{1}{N_v}\sum_{t=1}^{N_v} A_v^h}{\frac{1}{N_t}\sum_{t=1}^{N_t} A_t^h}, \tag{6}$$

where we respectively denote A_t^h and A_v^h as the text and vision output of the h-th attention head. A higher value of ρ^h indicating greater engagement of h-th attention head in vision question answering.

As shown in Fig. 5, we surprisingly find that most of attention heads don't answer actively, except the first layer. Thus, to further prune the visual computation in MLLMs, we propose to drop the attention heads of which $\rho < \alpha$, where α is the threshold to decide whether the attention head is active in visual processing.

3.4. Your model only needs text for deeper layers

As depicted in Fig. 3, compared with the first several layers, the cross-modal attention weights, *i.e.*, $Q_t K_v^T$, are significantly reduced in deeper layers, indicating a diminished influence of visual information on the text.

To further support this observation, we experimented with the LLaVA 1.5-13B model by selectively disabling visual computation in different sets of consecutive transformer blocks, each with a fixed length of 5 layers. For example, we skipped the computation of attention weights between visual and text tokens from layers 5 to 10 and evaluated the pruned model without fine-tuning on various visual question-answering benchmarks in a zero-shot setting. As reported in Tab. 1, the performance drop on different benchmarks decreases as we increase the depth of the pruned blocks. Notably, the model retains nearly the same performance when we skip cross-modal computation between layers 20 and 40, which is also consistent with Fig. 3.

Given that the output of MLLMs is ultimately text, this observation motivates us to directly skip all visual related computation in later layers, thereby reducing computational overhead. Specifically, for layers $l > L - N$, we propose to omit visual-related computations, including both visual and cross-modal attention, allowing the attention computation to be simplified as follows:

$$\text{Attention}(\mathbf{Q}_t, \mathbf{K}_t, \mathbf{V}_t) = \text{Softmax}\left(\frac{\mathbf{Q}_t\mathbf{K}_t^T}{\sqrt{d}}\right)\mathbf{V}_t, \quad (7)$$

With this design, only the text represented by $\mathbf{H}_T^{(l)}$ are processed independently in the last N layers.

4. Evaluations

4.1. Experiment setup

Studied models. In this work, we mainly focus on pruning visual computation in LLaVA models [38], specifically the LLaVA-1.5-7B and LLaVA-1.5-13B models. The hidden dimensions for LLaVA-1.5-7B and LLaVA-1.5-13B are 4096 and 5120, respectively, with intermediate sizes in the FFN of 11008 and 13824. The LLaVA-1.5-7B model has 32 layers, each containing 32 attention heads, while the LLaVA-1.5-13B model has 40 layers, each with 40 attention heads. Both models share a vocabulary size of 32,000. For efficiency comparison, we also include MoE-LLaVA-1.6Bx4 and MoE-LLaVA-2.7B [34], of which backbones are the StableLM-1.6B [4] and Phi-2.7B [25], respectively.

Simultaneously, we apply our proposed pruning strategies to Qwen2-VL-7B [3] and InternVL-2.0 [10] models to further demonstrate the pervasive nature of visual computation redundancy in current MLLMs.

Selected baselines. We select four methods as our baselines, including PruMerge+ [43], PyramidDrop [56], SparseVLM [56], and FastV [8]. Among these baselines, PruMerge+ and PyramidDrop require a fine-tuning step to accommodate the reduction of visual tokens, while SparseVLM and FastV are training-free methods designed to accelerate the model.

Evaluation benchmarks. We evaluate our method on diverse benchmarks covering visual question answering, scientific reasoning, and multimodal understanding. VQAv2 [20] tests visual and commonsense reasoning with open-ended questions on images. ScienceQA [40] includes multi-modal questions on science topics, and we focus on the SQA^I subset, which contains questions that specifically include images as part of the question context. TextVQA [46] challenges models with text-recognition questions on different images. GQA [24] tests fine-grained visual reasoning with multistep question-answer pairs. POPE [30] measures object hallucination under varying conditions. MME [19] and MMBench [39] assess multimodal reasoning across thousands of image-text pairs. We show the number of examples in each selected benchmark for model evaluation in Tab. 2. These benchmarks offer a comprehensive evaluation of our method in diverse task domains.

Implementations. In our experiments, we apply individual and combined pruning strategies to reduce visual computation redundancy in the model. We evaluate our approach in two settings: training-based and training-free. In the training-based setting, we use the same data and training protocol as the public repository of LLaVA-1.5 models, training from scratch with reduced visual computation redundancy. The number of iterations remains the same to allow a fair comparison with the original model. In the training-free setting, we directly apply different pruning strategies to studied models, *i.e.*, Qwen2-VL-7B, and InternVL-2.0, without any fine-tuning. All experiments were conducted on a single node with 8 A100-80G GPUs, with detailed results provided in Appendix A.

4.2. Comparison with efficient LLaVA models

We begin by evaluating the effectiveness of our proposed method for pruning the LLaVA-1.5-7B and LLaVA-1.5-13B models. By combining our four strategies, we reduce the FLOPs to 25% and 12% of the original LLaVA model, respectively. The results are reported in Tab. 2.

With the same computational budget, *i.e.*, the same FLOPs, our pruning method consistently achieves the best results on the four benchmarks, showing an average performance gap of 3.7%, 1.1%, 2.2%, 0.45% over the runner-up method on GQA, VQAv2, POPE, and MMB, respectively. Although our method does not perform best on certain test cases, such as LLavA-7B on SQA^I and TextVQA, our pruned model achieves comparable results, with only a minor average performance drop of 0.5%.

It should be noted that our pruned model, using only 12% of parameters for visual computation, can achieve performance comparable to selected baselines with a computation budget approximately 2× larger. Specifically, on the largest dataset, GQA, our 7B model surpasses the runner-up method with a clear gap of 3.4%.

Furthermore, when comparing the results with MoE-LLAVA-1.6Bx4 and MoE-LLaVA-2.7Bx4—models with a total number of parameters similar to LLaVA-7B and LLaVA-13B, respectively—our pruned model demonstrates greater efficiency and superior performance on selected benchmarks. While MoE is often utilized to scale up model parameters while improving inference efficiency, given the challenges of MoE training, this comparison suggests that pruning from a large dense model is also a promising, practical, and simple approach to retain most of the performance while significantly reducing the computation budget, making it more suitable for efficient deployment.

4.3. Pruning with different granularity

To demonstrate the scalability of our method in pruning visual computation redundancy, we compare our proposed strategy with PyramidDrop and FastV at different pruning granularities on the two largest benchmarks, VQAv2 and GQA. Scores on these benchmarks are reported along with corresponding FLOPs, as shown in Fig. 6.

Table 2. Performance comparison with various efficient MLLMs on different benchmarks. The '-' symbol indicates missing values where certain evaluations were not available. **We list the number of test examples in each benchmark.**

Backbone	Model	FLOPs (T)	GQA 12K	VQAv2 107K	SQA^I 4K	TextVQA 5K	POPE 9K	MME 2K	MMB$_{en}$ 4K
StableLM	MoE-LLaVA-1.6Bx4	1.96	60.4	–	62.6	47.8	84.3	1300.8	59.4
Phi	MoE-LLaVA-2.7Bx4	5.42	61.1	–	68.7	50.2	85.0	1396.4	65.5
LLaVA-1.5-7B	Original	7.63	62.3	79.2	69.4	58.1	87.3	1492.8	65.4
	PruMerge+ [43]	1.91	–	76.8	68.3	**57.1**	84.0	1462.4	64.9
	Pyramid-Drop [56]	1.91	57.3	75.1	**69.3**	55.9	83.4	1300.8	59.4
	SparseVLM [64]	1.91	50.2	62.9	67.1	52.5	72.0	1090.8	60.0
	FastV [8]	1.91	56.5	73.5	69.1	57.5	77.8	1380.2	62.9
	Ours	1.91	**61.6**	**78.0**	69.0	56.3	**86.8**	**1483.6**	**65.5**
		0.92	60.7	77.4	68.0	55.2	86.6	1429.6	64.1
LLaVA-1.5-13B	Original	14.89	63.0	79.7	72.4	59.8	86.4	1558.0	68.7
	PruMerge+ [43]	3.72	–	77.8	71.0	**58.6**	84.4	1485.0	65.7
	Pyramid-Drop [56]	3.72	58.4	76.7	**72.3**	58.1	83.3	**1506.6**	65.8
	SparseVLM [64]	3.72	47.2	57.3	–	53.5	65.5	1090.8	63.1
	FastV [8]	3.72	59.4	77.1	71.2	58.5	82.0	1506.8	67.1
	Ours	3.72	**62.5**	**78.9**	71.3	58.3	**86.3**	1457.3	**67.4**
		1.79	61.2	78.1	70.7	56.7	86.4	1426.2	65.5

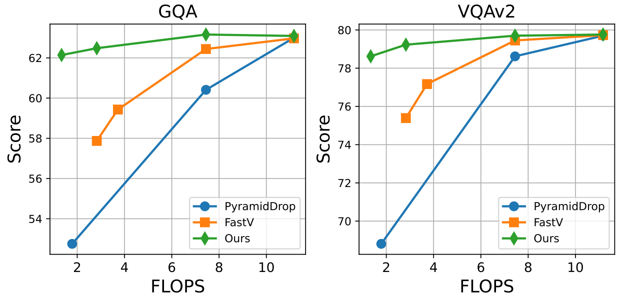


Figure 6. Evaluation results of the pruned LLaVA-1.5-13B using various strategies with differing pruning granularities. Results are reported on the two largest benchmarks: GQA and VQAv2.

It can be observed that as FLOPs for visual computation decrease, the performance of the pruned model also declines. Specifically, reducing FLOPs from 75% to 19% led to a performance drop from 71.35% to 66.63% on average across the two benchmarks for the model pruned using FastV. In contrast, rather than pruning tokens, our approach targets redundant visual computations at the parameter and computation pattern levels, resulting in only a 0.5% performance decrease. These results further support our claim that a substantial amount of visual computation redundancy can be effectively pruned in current MLLMs.

4.4. Computational redundancy beyond LLaVA

The visual computational redundancy is not unique to LLaVA. To validate the broader applicability of our pruning strategy, we applied it to additional MLLMs, including Qwen2-VL-7B and InternVL-2.0, without fine-tuning due to limited training data. We evaluated performance on GQA and POPE benchmarks, adjusting pruning granularity to match original model performance with minimal FLOPs. As shown in Fig. 7, we observe that, even without fine-tuning, pruning visual computations in these models at suitable ratios does not compromise their performance. Furthermore, larger MLLMs appear to accommodate higher pruning ratios, as evidenced by the results from pruning InternVL-2.0 at various model scales. More details can be found in Appendix C.

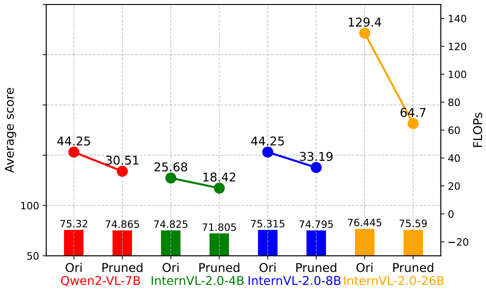


Figure 7. Evaluation results of the pruned Qwen2-VL-7B, InernVL-2.0-4B/8B/26B. The results are averaging on GQA and POPE.

4.5. Discussion

On the neighbor-aware visual attention. We vary the number of neighboring visual tokens in the attention computation by setting the radius from 1 to 13 in increments of 2, and report the results in Tab. 3. The peak performance occurs at a radius of 7, which aligns with the observation in Fig. 2, where most non-sparse attention values fall within this spatial range. This suggests that carefully selecting the optimal radius for neighboring token attention computation can accelerate both the training and inference of our model.

On the non-active visual attention head dropping. In our method, we use the ratio between visual attention and text attention to identify non-active attention heads for visual computation. To further study the effect of dropping attention heads on model performance, we vary the number of randomly selected attention heads dropped for visual computation in each layer from 5 to 35, and report the results in Tab. 4. The results show that dropping attention heads within this range mostly preserves the model's original performance. Generally, as the number of dropped attention heads increases, model performance gradually declines. Notably, even with 35 attention heads dropped, there is only a 2.0% average performance drop, indicating significant redundancy in visual computation within the model.

On the sparse visual projection in FFN. Sparse projection is employed to better leverage the sparsity of the visual representation after significantly pruning the model's visual computation. As shown in Tab. 2, we increase the ratio of dropped neurons in the FFN from 25% to 50%, thereby reducing the FLOPs from 1.91 to 0.92 on LLaVA-7B, and from 3.72 to 1.79 on LLaVA-13B. Without applying sparse projection in the FFN and using the same pruning strategy as in Tab. 2, the LLaVA-13B model achieves a performance of 62.5 on GQA and 79.2 on VQAv2, which is quite similar to the model with 25% dropped neurons in the FFN. These results suggest, on one hand, the redundancy in visual computation within the FFN. On the other hand, we observe that although increasing the drop ratio results in further performance decline, it still outperforms models with similar FLOPs, *i.e.*, the MoE-LLaVA models.

On the layer-dropping for visual computation. Layer dropping is the most effective strategy to reduce FLOPs, where only text is processed within the dropped layers. While we previously provided results for block-wise layer dropping, here we conduct further experiments by continuously dropping layers for visual computation. As shown in Tab. 5, even when the last 20 layers are dropped, the model's performance remains almost unchanged compared to the original model. This strongly suggests that the visual computation is extremely redundant in deep layers, leaving great potential to prune for improving the efficiency.

Why not directly prune the parameters for both the vision and texts? We address the redundancy in visual token computations, reducing their overhead while preserving text token computations. To explore if text tokens hold similar redundancy, we ran an experiment pruning 20 attention heads for only visual tokens versus both vision and text tokens. Without fine-tuning, pruning just visual tokens resulted in an average performance of 67.1% across VQAv2, GQA, SQA, and TextVQA, while pruning both led to a drastic drop to 4.3%. This indicates much higher redundancy in visual computations than in text within current MLLMs.

Table 3. Results of applying the neighbor-aware attention mechanism as a visual pruning strategy to the LLaVA-1.5-13B model. The 'radius' parameter defines the scope of neighboring tokens considered in visual token computation.

Radius	GQA	VQAv2	SQA	TextVQA	Avg
1	62.8	79.1	71.2	58.5	67.9
3	62.7	79.2	71.5	58.5	68.0
5	62.8	79.7	70.9	59.9	68.3
7	63.2	79.8	71.7	59.6	68.6
9	63.2	79.7	71.3	59.8	68.5
11	63.0	79.6	70.2	59.5	68.1
13	62.8	79.5	71.6	59.2	68.3

Table 4. Results of dropping non-active attention heads as a visual pruning strategy to prune the LLaVA-1.5-13B model. '# Heads' indicates the number of dropped non-active attention heads per layer used in visual token computation.

# Heads	GQA	VQAv2	SQA	TextVQA	Avg
5	62.7	79.7	71.3	59.6	68.3
10	62.5	79.4	72.0	59.9	68.4
15	62.5	79.1	71.6	58.1	67.9
20	62.0	78.9	72.0	58.4	67.8
25	62.1	78.9	70.0	58.6	67.4
30	62.0	78.5	71.2	57.5	67.3
35	61.0	77.7	71.1	55.2	66.3

Table 5. Results of applying the layer-dropping strategy to prune the LLaVA-1.5-13B model.# layer indicate the number of last several layers to drop for visual-related computation.

# Layer	GQA	VQAv2	SQA	TextVQA	Avg
5	63.1	79.8	71.9	59.1	68.5
10	63.0	79.7	72.3	59.0	68.5
15	63.0	79.8	71.6	59.6	68.5
20	63.1	79.7	71.1	58.1	68.1
25	62.2	79.0	71.5	55.5	67.1
30	61.5	78.0	69.5	56.3	66.4
35	55.0	72.4	71.6	53.6	63.2

Efficiency analysis on token pruning and computation pattern pruning. We provide a comparison of efficiency across various methods with differing numbers of input visual tokens in Fig. 8. The results indicate that, in contrast to token pruning-based approaches, addressing visual computational redundancy at the computational pattern level yields a greater efficiency advantage for long visual sequences. This approach effectively mitigates the escalating computational overhead associated with handling large numbers of visual tokens, demonstrating its superior scalability in processing extended visual sequences.

5. Conclusion

In this work, we address the challenge of pruning MLLMs for efficient computation. Unlike text, visual information is sparse and redundant. Prior work has focused on reducing visual tokens; we instead analyze redundancy in parameters

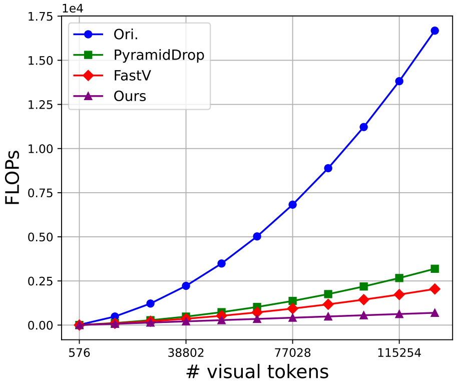


Figure 8. Comparison of the FLOPs for different methods varying the number of visual tokens. The studied model is LLavA-1.5-13B.

and computation patterns. Our strategies—neighbor-aware visual attention, non-active visual heads dropping, sparse visual projection in FFNs, and layer dropping—reduce LLaVA's computational overhead by 88% while largely preserving performance. Additional experiments on Qwen2-VL-7B and InternVL-2.0 further confirm that visual computation redundancy is prevalent across MLLMs.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 1

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 6

[4] Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, et al. Stable lm 2 1.6 b technical report. *arXiv preprint arXiv:2402.17834*, 2024. 6

[5] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021. 1

[6] Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for a lite vision-language model. *arXiv preprint arXiv:2402.11684*, 2024.

[7] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023. 1

[8] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 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, 2025. 2, 6, 7

[9] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 1

[10] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 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, 2024. 6

[11] Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*, 2023. 2

[12] Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, et al. Mobilevlm v2: Faster and stronger baseline for vision language model. *arXiv preprint arXiv:2402.03766*, 2024. 2

[13] Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*, 2023. 1

[14] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, et al. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024. 1

[15] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[16] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019. 3

[17] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023. 2

[18] Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337. PMLR, 2023. 3

[19] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 6

[20] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[21] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 3

[22] Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. Efficient multimodal learning from data-centric perspective. *arXiv preprint arXiv:2402.11530*, 2024. 2

[23] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290, 2024. 1

[24] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[25] Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 2023. 2, 6

[26] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. 1

[27] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 1

[28] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 1

[29] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *arXiv preprint arXiv:2407.02392*, 2024. 2

[30] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305, Singapore, 2023. Association for Computational Linguistics. 6

[31] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024. 2

[32] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024. 2

[33] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 1

[34] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Jinfa Huang, Junwu Zhang, Yatian Pang, Munan Ning, et al. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024. 6

[35] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024. 2

[36] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 1

[37] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 2

[38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1, 6

[39] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023. 6

[40] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 6

[41] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024. 1

[42] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019. 3

[43] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024. 2, 6, 7

[44] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024. 2

[45] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 1

[46] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019. 6

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

[48] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1

[49] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024. 1

[50] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578, 2024. 1

[51] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[52] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 3

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

[54] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023. 1

[55] Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, En Yu, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Small language model meets with reinforced vision vocabulary. *arXiv preprint arXiv:2401.12503*, 2024. 2

[56] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024. 2, 6, 7

[57] Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, Maosong Sun, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. *arXiv preprint arXiv:2403.11703*, 2024. 2

[58] Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, Maosong Sun, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. *arXiv preprint arXiv:2403.11703*, 2024. 2

[59] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 2

[60] Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13807–13816, 2024. 1

[61] Ya-Qi Yu, Minghui Liao, Jihao Wu, Yongxin Liao, Xiaoyu Zheng, and Wei Zeng. Texthawk: Exploring efficient fine-grained perception of multimodal large language models. *arXiv preprint arXiv:2404.09204*, 2024. 2

[62] Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. Tinychart: Efficient chart understanding with visual token merging and program-of-thoughts learning. *arXiv preprint arXiv:2404.16635*, 2024. 2

[63] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023. 1

[64] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024. 2, 7

[65] Zeliang Zhang, Xiaodong Liu, Hao Cheng, Chenliang Xu, and Jianfeng Gao. Diversifying the expert knowledge for task-agnostic pruning in sparse mixture-of-experts. *arXiv preprint arXiv:2407.09590*, 2024. 2

[66] Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*, 2024. 2

[67] Yichen Zhu, Minjie Zhu, Ning Liu, Zhicai Ou, Xiaofeng Mou, and Jian Tang. Llava-phi: Efficient multi-modal assistant with small language model. *arXiv preprint arXiv:2401.02330*, 2024. 2

Appendix

A. Experiment details

For the LLaVA pruning with training. We employed training data and settings identical to those used in LLaVA-1.5. All experiments were conducted on a single node equipped with 8 A100-80G GPUs. The pre-training phase took 4 hours, while instruction fine-tuning took 20 hours.

For the MLLMs pruning without training. For the experiments of pruning models without the training for performance restoration, we use one A100-80GB GPU.

Pruning details for our models. In Tab. 2, we present the results of the LLaVA pruned by the combination of our strategies. Here, we present the pruning details in Tab. 6.

B. More results on LLaVA-1.5-7B

We provide a detailed evaluation of pruning LLaVA-1.5-7B using various strategies, including neighbor-aware attention computation (Tab. 7), dropping inactive attention heads (Tab. 8), and layer dropping (Tab. 9). The results show that a radius of 5 achieves efficient attention computation with strong performance. Additionally, dropping 12 attention heads and the last 12 layers during visual computation allows for significant acceleration without compromising performance.

C. More results on Qwen2-VL-8B and InternVL-2.0-4B/8B/26B

We present more results of pruning the Qwen-2-VL-8B and InternVL-2.0-4B/8B/26B without the fine-tuning under different pruning strategies and granularity.

Pruning details. For Qwen-2-VL-8B, we respectively use the non-active attention head dropping and neighbor-aware attention computation to reduce the FLOPs to 75%. For InternVL-2.0-4B/8B, we set the radius for attention computation as 4 to reduce the FLOPs to 75%, and integrate the non-active attention head dropping with 24 attention heads maintaining to further reduce to 66%, then discard the visual-related computation after 24 layers to reduce to 50%, while changing the layer from 24 to 16 reduces the FLOPs to 44%. For InternVL-2.0-8B, we further reduce the last layer for visual computation to 12 to reduce the FLOPs to 33%. For InternVL-2.0-26B, by setting the radius for visual attention computation as 4, maintaining 24 active visual attention heads, and discarding the visual-related computation after the 36-th layer, we can reduce 50% FLOPs, while changing the discarding layer to 24 and 19 can further reduce to 33% and 25% FLOPs, respectively. The results are shown in Fig. 9 and Fig. 10

Table 6. Summary of LLaVA-ours models with corresponding pruning parameters. We denote # Heads as the number of attention heads retained for the visual attention computation, and Radius as the radius of neighbor visual tokens for attention computation. The # Layers is denoted as the number of last layers to drop for the visual-related computation. The # Neurons is denoted as the ratio of neurons retained at the FFN layer.

Model	**FLOPs**	**# Heads**	**Radius**	**# Layers**	**# Neurons**
LLaVA-7B-ours	1.91	24	5	16	50%
	0.92	16			25%
LLaVA-13B-ours	3.72	30	5	20	50%
	1.79	20			25%

Table 7. Results of applying the neighbor-aware attention mechanism as a visual pruning strategy to the LLaVA-1.5-7B model. The 'radius' parameter defines the scope of neighboring tokens considered in visual token computation.

Radius	GQA	VQAv2	SQA	TextVQA	Avg
1	61.8	78.6	69.9	56.9	66.8
3	61.9	78.8	69.4	57.3	66.9
5	62.2	78.9	69.0	57.9	67.0
7	61.9	79.0	69.1	58.6	67.1
9	62.2	79.0	68.7	58.8	67.2
11	60.6	78.2	69.2	57.6	66.4

Table 8. Results of dropping non-active attention heads as a visual pruning strategy to prune the LLaVA-1.5-7B model. '# Heads' indicates the number of dropped non-active attention heads per layer used in visual token computation.

# Heads	GQA	VQAv2	SQA	TextVQA	Avg
4	62.1	78.9	68.0	58.3	66.8
12	61.9	78.6	68.3	57.3	66.5
20	61.6	78.1	67.7	57.2	66.1
28	60.8	77.1	67.7	56.0	65.4

Table 9. Results of applying the layer-dropping strategy to prune the LLaVA-1.5-13B model. # layers indicates the number of last several layers to drop for visual-related computation.

# Layers	GQA	VQAv2	SQA	TextVQA	Avg
4	62.4	79.1	68.7	57.8	67.0
8	62.2	79.1	68.7	58.3	67.0
12	62.2	79.0	68.9	57.5	67.0
16	62.1	78.7	68.4	57.0	66.6
20	61.4	78.0	68.0	55.3	65.7
24	59.3	75.4	68.1	54.0	64.2
28	54.5	71.1	68.4	51.6	61.4

It can be shown that even without any fine-tuning, our proposed pruning strategies can efficiently reduce the required FLOPs with remaining similar performance compared with the original model. It should be noticed that from the results of pruning InternVL-2.0 models, we can see that *a larger model usually has a larger computation redundancy.* For example, while the use of 44% FLOPs causes a large performance drop for InternVL-2.0-8B model, the InternVL-2.0-26B model still has a comparable performance compared

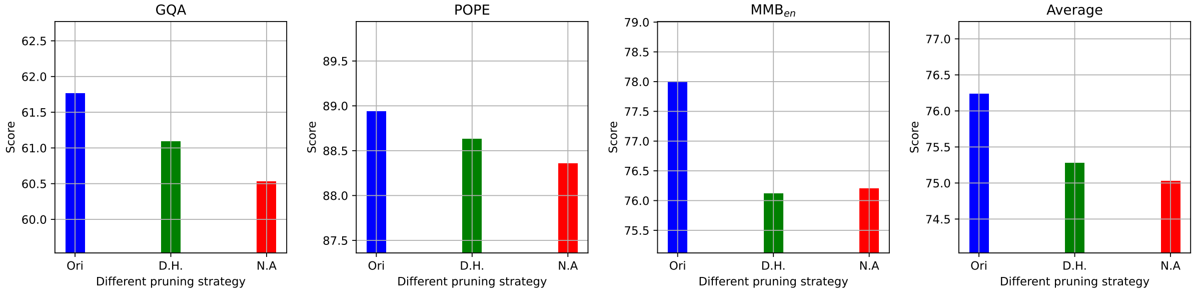


Figure 9. Evaluation results of Qwen2-VL-8B under different pruning strategies. We denote D.H. as dropping non-active visual attention heads during the visual-related computation, and N.A. as the neighbor-aware visual attention computation.

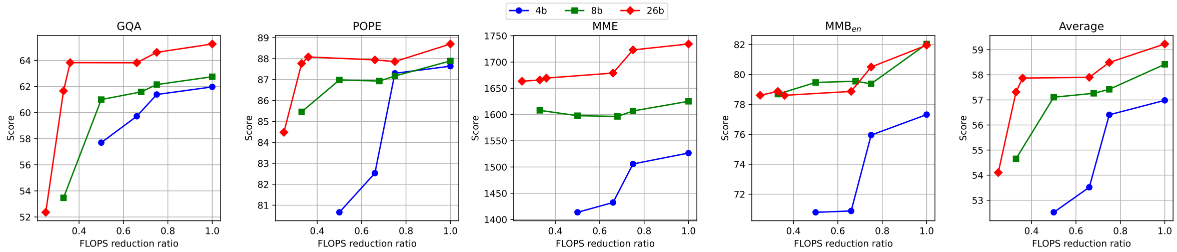


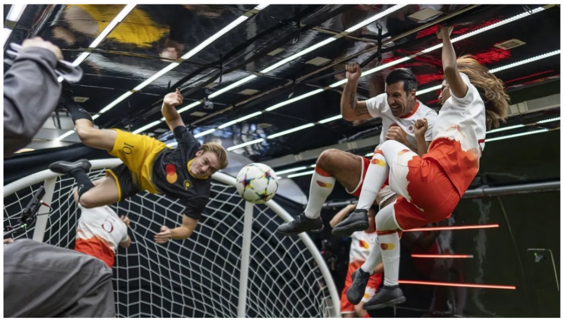
Figure 10. Evaluation results of InternVL-2.0-4B/8B/26B under different pruning ratios. The ratio of 1 indicates the original model without pruning.

with the original one. Also, for a given pruning ratio, we should carefully select the pruning strategy. From the results of Qwen2-VL-8B, the use of neighbor-aware attention computation could cause a larger performance drop than the use of non-active attention head dropping. It remains further work to study how to select a better combination of pruning strategies to get a good performance with improved efficiency.

D. Demo application

We also present three real-world examples to show the performance of the model (LLaVA-1.5-13B) before and after pruning. The results are depicted in Fig. 11, Fig. 12, and Fig. 13. We highlight the expected keywords in violet. Even with only 12% of FLOPs allocated to visual-related computation, MLLM demonstrates the ability to accurately capture semantic information. This highlights the significant potential of MLLM visual computation pruning in accelerating real-world applications.

Q: What is uncommon in the image?



LLaVA-1.5-13b

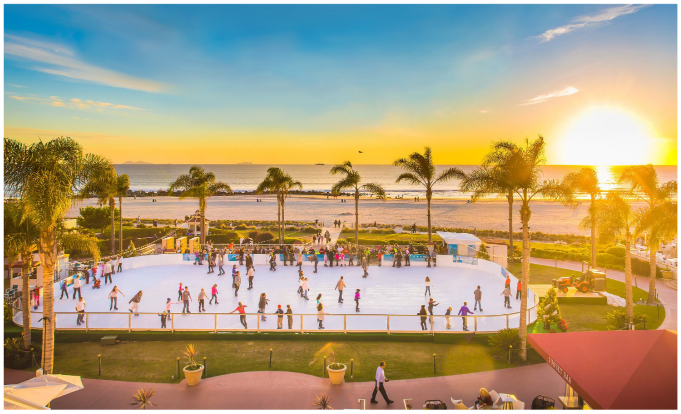
A: The uncommon aspect of the image is that the three men are playing soccer while hanging from a net. This is not a typical way to play soccer, as the sport is usually played on a field with a goal at each end. The men are engaging in a more unconventional and challenging form of the game, which adds an element of excitement and novelty to the scene.

After pruning (12% FLOPs)

A: In the image, there is a group of people playing soccer in a stadium. The uncommon aspect of this scene is that the players are not only playing soccer, but they are also performing acrobatics and doing stunts, such as jumping and diving. This is unusual because soccer is typically played as a sport, and acrobatics and stunts are not typically associated with the game.

Figure 11. Demo application result #1.

Q: Please describe the image as detailed as possible



LLaVA-1.5-13b

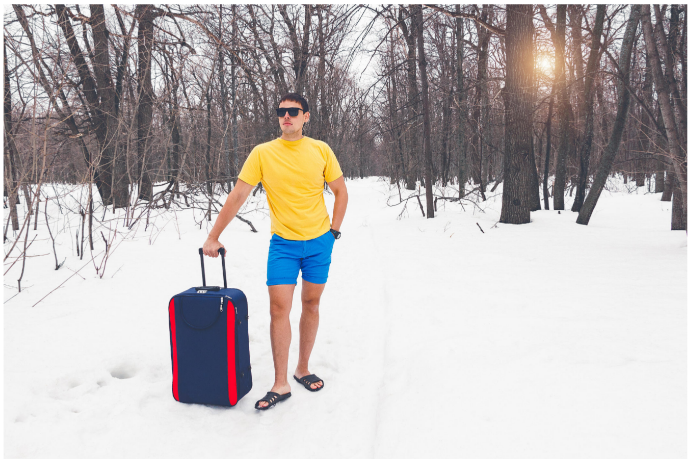
A: The image depicts a lively outdoor ice skating rink situated near a beach. There are numerous people of various ages enjoying the rink, with some skating and others watching. The ice rink is surrounded by palm trees, adding a tropical touch to the scene. In addition to the skaters, there are several chairs scattered around the area, likely for spectators or resting skaters. The beach is visible in the background, with the ocean providing a picturesque backdrop for the ice skating activity.

After pruning (12% FLOPs)

A: The image captures a lively scene at a beach resort, where a large group of people are gathered around a skating rink. The rink is filled with skaters, some of whom are performing tricks and enjoying the activity. The atmosphere is vibrant and energetic, with people of various ages and skill levels participating in the fun. In the background, the ocean can be seen, adding to the picturesque setting of the resort. The sun is setting, casting a warm glow over the scene, and creating a beautiful backdrop for the skaters and spectators. The resort appears to be a popular destination for both skaters and onlookers, as the rink is bustling with activity.

Figure 12. Demo application result #2.

Q: Please describe the image as detailed as possible



LLaVA-1.5-13b

A: The image features a man standing in the snow, wearing a yellow shirt and blue shorts. He is holding a red suitcase, possibly preparing for a trip or just returning from one. The man is positioned in the center of the scene, and the snowy surroundings create a wintry atmosphere.

After pruning (12% FLOPs)

A: The image features a man standing in the snow, wearing a yellow shirt and blue shorts. He is holding a red suitcase, which is placed on the ground beside him. The man appears to be posing for a picture in the snowy landscape. In the background, there are several trees, adding to the wintery atmosphere. The scene captures the man's adventurous spirit as he embraces the cold weather and enjoys his time outdoors.

Figure 13. Demo application result #3.