An Image is Worth 1/2 Tokens After Layer 2: Plug-and-Play Inference Acceleration for Large Vision-Language Models

Liang Chen[1], Haozhe Zhao[1], Tianyu Liu[2], Shuai Bai[2], Junyang Lin[2]
Chang Zhou[2], and Baobao Chang[1†]

[1] National Key Laboratory for Multimedia Information Processing, Peking University
[2] Alibaba Group
leo.liang.chen@outlook.com, chbb@pku.edu.cn

Abstract. In this study, we identify the inefficient attention phenomena in Large Vision-Language Models (LVLMs), notably within prominent models like LLaVA-1.5, QwenVL-Chat, and Video-LLaVA. We find that the attention computation over visual tokens is extremely inefficient in the deep layers of popular LVLMs, suggesting a need for a sparser approach compared to textual data handling. To this end, we introduce FastV, a versatile plug-and-play method designed to optimize computational efficiency by learning adaptive attention patterns in early layers and pruning visual tokens in subsequent ones. Our evaluations demonstrate FastV's ability to dramatically reduce computational costs (e.g., a 45% reduction in FLOPs for LLaVA-1.5-13B) without sacrificing performance in a wide range of image and video understanding tasks. The computational efficiency and performance trade-off of FastV are highly customizable and Pareto-efficient. It can compress the FLOPs of a 13B-parameter model to achieve a lower cost than that of a 7B-parameter model while still maintaining superior performance. We believe FastV has practical value for the deployment of LVLMs in edge devices and commercial models. Code is released at github.com/pkunlp-icler/FastV.

Keywords: Large Vision-Language Model · Inference Acceleration

1 Introduction

Large Vision-Language Models (LVLMs) have become a hit in both computer vision and natural language processing studies. We have witnessed tremendous creative research and applications that are built upon powerful LVLMs [2, 23, 27, 36]. From describing the given picture to navigating the internet [48], using smartphones [39] and making decisions in the real world [6, 9], large language models with vision abilities are reshaping how we interact with AI systems, which cannot be achieved solely by language or vision uni-modal models.

† Corresponding author.

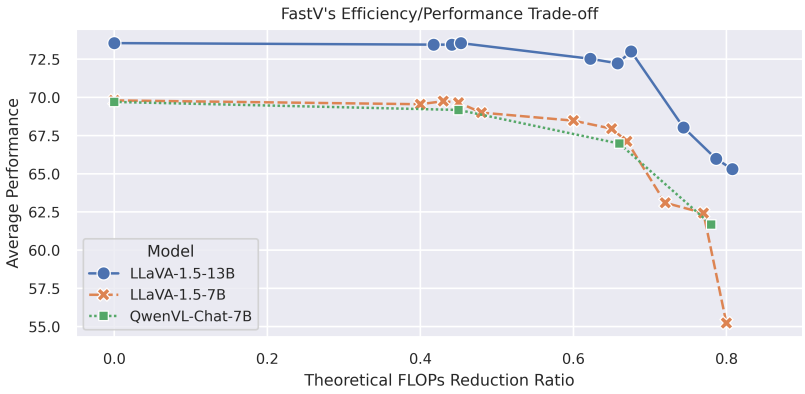

Fig. 1: The Efficiency/Performance trade-off curve of FastV. The x-axis stands for the theoretical FLOPs reduction ratio under different FastV configurations. The y-axis stands for performance under different settings, we report the average scores of {Nocaps (Cider), Flickr30k (Cider), A-OKVQA (Acc), MMMU (Acc)}. We can see that FastV can achieve 45% FLOPs reduction with nearly no performance loss for different models.

Currently, a majority of popular LVLMs rely on sequential visual representation, where images are transformed into hundreds or thousands of tokens when feeding them to LLM along with language prompts [2, 27, 32, 47, 49]. As LVLMs leverage the advanced emergent capabilities inherent in their language components, they concurrently face a surge in computational complexity, correlating with cost increments. This complexity stems from the principle that the proficiency of Large Language Models (LLMs) is predominantly influenced by their scale. Two critical areas remain under-explored in this context: 1) How do language models process and interpret images? and 2) While the efficient training and inference of LLMs have attracted considerable attention, these dimensions within LVLMs are yet to be thoroughly examined and understood.

In this paper, we uncover the fact that current LVLMs actually apply an inefficient way while processing image information. Specifically, the image tokens receive strikingly lower attention scores compared to their textual counterparts within the token-based LVLMs like LLaVA. The degree of imbalance also varies between the shallow and deep layers. In the image captioning tasks, we observed that within the deep layers (after layer 2) of renowned LVLMs such as LLaVA 1.5, image tokens garner an average attention score that amounts to only 0.21% of the score attributed to system prompts. In contrast, this figure reaches 50% in the initial two layers. These observations raise questions upon the optimal utilization of visual information within LVLMs.

To address the problem, we assume a plausible explanation is that the high redundancy in visual signals leads to the aggregation of image-related, instruction-specific features onto certain "anchor" tokens through the self-attention mechanism in the shallow layers. Notably, these anchor tokens are not image tokens.

In deep layers, attentions are focused on those anchor tokens, leading to significantly reduced attention on the image tokens themselves.

The phenomena inspires to propose FastV, a dynamic image tokens pruning method to reduce the inference cost of LVLMs. Our findings suggest an intriguing possibility: Given that image tokens contribute minimally to output generation in deeper layers due to diminished attention, why not consider removing them at these stages? FastV implements an image token pruning strategy at one specific layer of LLM. Prior to this layer, computations proceed as usual. Beyond this selected layer, image tokens are re-evaluated based on their average received attention scores. Tokens falling below a predefined attention score threshold are then selectively discarded in subsequent layers, streamlining the process by focusing on the most impactful tokens.

Compared to other attention-based methods for accelerating inference, such as sparse attention, FastV's most notable distinction lies in its direct elimination of tokens. This approach not only bypasses the computational demand of the self-attention module but also the Feed-Forward Network (FFN) module in deeper layers. As a result, FastV achieves a great theoretical reduction in FLOPs while maintaining relatively high performance as shown in Figure 1's experiment on LLaVA and Qwen-VL-Chat models. Our experiment on LLaVA-1.5-13B model shows that we can filter out 50% image tokens after layer 2 without sacrificing the average performance on a combination of Vision-Language tasks including captioning tasks like Nocaps [1], Flickr30K [33], multimple choice tasks like A-OKVQA [35], MMMU [46], complex embodied reasoning task like PCA-Bench [5, 6], tasks requiring detailed OCR ablitily like OCR-VQA [31], more challenging video understanding tasks [12, 43, 44] and more fine-grained evaluation like MME [10], MMVet [45] and SeedBench [16]. Our latency test experiment on A-OKVQA showed that LLaVA-13B model with FastV could achieve a lower latency than LLaVA-7B model while maintaining superior performance. This result highlights the effectiveness of FastV in balancing the trade-off between speed and accuracy in LVLMs.

Researches [20, 27] underscore the significance of enhancing image resolution for the performance of LVLMs. However, it's equally important to note that increased resolution comes with its own challenges, including a rise in the computational costs such as longer image token sequence and inference latency. We also conduct experiments on training LVLM in different image feature resolution by setting pooling layer of different strides. Specifically, with an equal number of image tokens, models equipped with FastV can process higher resolution images, leading to better performance than models limited to lower resolution features. This finding highlights the potential to enhance downstream performance by increasing image resolution without incurring additional inference costs.

In summary, the contribution of the work are three-folds:

1. Identify and analyze the inefficient visual attention phenomena in prevailing LVLMs.
2. Propose FastV, a plug-and-play method to significantly reduce inference cost for LVLMs without sacrificing performance inspired by our observation.

3. Validate the effectiveness of FastV on a wide range of vision-language tasks across different LVLMs with thorough ablations.

2 Related Work

Large Vision-Language Model. To benefit from the advancement of LLM and integrate visual information into the LLM, large Vision-Language Models utilize a Visual Prompt Generator [17] to transform the visual embeddings into prompts that the language model can comprehend [18, 27], resulting in a significant increase in required tokens. Handling higher resolution images inevitably necessitates a quadratic increase in the number of needed tokens. For instance, LLAVA process 336x336 images into 576 tokens [25] and process images with a greater resolution of 672x672 into 2304 tokens [26]. Fuyu [3], in a similar vein, translates pixel-level images of 1080x1080 into 1296 tokens. Understanding and generating multiple images or videos also inherently demands an escalated count of tokens for vision information. Both Video-Poet [13] and Unified-IO2 [28] are compelled to reserve thousands of tokens within the context to facilitate the understanding and generation of multiple images or videos. Large multimodal models like Gemini [36] and LWM [23] highlights the significance of long context in developing a robust understanding of the world model and extending the context length to 1M to address the issue of escalating context requirements.

Inference Optimization for LLM. Efficient inference in LLMs is challenged by their autoregressive generation where each token prediction depends on the preceding context. Hence, considering the quadratic complexity of computation's attention during training, as the context length increases, the generation becomes progressively slower. To tackle these challenges, pioneering studies fall into two categories: methods optimizing memory consumption for attention module like FlashAttention, vLLM and RingAttention [7, 8, 15, 24], which ensure no drastic shifts in the results, and methods like StreamingLLM and FastGen [11, 41] that simplify computations by pruning redundant attention computation. We are interested in the second kind of methods since they are proposed inspired by the distinct attention patterns observed in LLM's inference. While these methods have boosted the inference efficiency of LLMs, they are designed for text-only language models, and whether their effectiveness can be transferred to LVLMs remain under-explored. There is previous work attempt to handle the long-context in LVLMs efficiently, like LLaMA-VID [19], which utilizes cross-attention to effectively represent each video frame with two key tokens, the requirement for an additional fine-tuning stage obstructs its broad applicability for different LVLMs.

Token Reduction for VLMs. There have been studies on improving efficiency for Vision-Language Models (VLMs) before the era of large vision-language models. A majority of them focus on token reduction for vision transformers (ViTs). Various methods, such as EViT [21], SPViT [14], and Pumer [4], have been proposed for ViTs. More recently, PYRA [42] has enhanced the training and inference of ViTs via a specialized token merging technique. FastV is the first

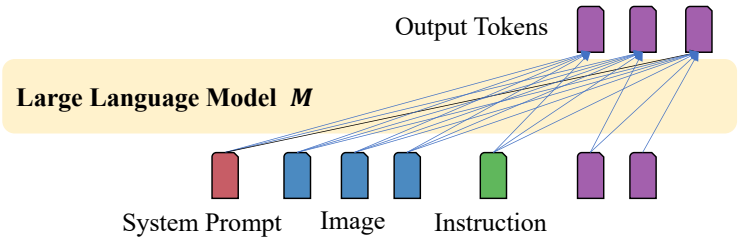

Fig. 2: Classic network architecture of LVLM. Image tokens and different types of text tokens are sent to the LLM as input. LLM generates output tokens conditioned on the input tokens and preceding output in an auto-regressive manner.

to explore visual token reduction for Large Vision-Language Models (LVLMs), which uses language as an interface for various vision-language tasks. FastV utilizes the signal from LLM to guide the pruning of visual tokens, a strategy not previously explored. We are the first to demonstrate the effectiveness of token reduction in video-QA and various comprehensive LVLM benchmarks. Another significant advantage of FastV over previous methods is its simplicity; it can be applied to any LVLM without requiring model retraining.

3 Inefficient Visual Attention in LVLMs

3.1 Preliminaries

In this section, we delve into how LVLMs process visual tokens during output generation from the perspective of self-attention module. For an image-question pair (d, t), the given LVLM M, usually in the structure of transformer [37] decoder, predicts the answer $\hat{y} = M(d, t)$ in an auto-regressive manner:

$$p(\hat{y}) = \prod_{i=1}^{N} p_M\left(\hat{y}_i \mid \hat{y}_{1\sim i-1}; d; t\right) \tag{1}$$

Multimodal information, encompassing both images and text, is transformed into sequential embeddings prior to being processed by the transformer model. For images, a commonly used approach is to employ a pretrained encoder, such as CLIP-VIT [34], to extract visual features. These features are then linearized by eliminating the spatial dimension. Additional linear transformations [25, 49] or cross-attention [2, 18] modules are utilized to adjust the size of the visual features to match the embedding size of the Large Language Model (LLM) and to achieve semantic alignment. Regarding text, a tokenizer breaks down the natural language into discrete tokens and then performs an embedding lookup to form text embeddings. In the rest of the paper, we refer to 'visual tokens' and 'text tokens' not merely as the discrete units of visual and textual data but as the embeddings derived from these units.

As illustrated in Figure 2, after preprocessing the image and text token to a unified embedding space, they are fed to the transformer decoder to generate output tokens. The input tokens at each decoding step can be categorized into four distinct types: system prompt (sys), image tokens (img), user instruction (ins), and output tokens (out). The system prompts for LVLMs usually inherit the backbone LLM, used as a general message to control the LLM's behavior, which is decided during the instruction tuning stage of LLM. Image tokens are the linearized image features transformed by a pretrained vision encoder. User instruction specifies the query question for the given image. Output tokens are generated step by step conditioned on the preceding tokens.

3.2 Experiment Settings

To explore how LVLMs process image tokens, we first randomly sample N image-text pairs $D = \{(d^1, t^1), ..., (d^N, t^N)\}$ from a combination of vision langauge tasks including image caption (Flickr30K), embodied reasoning (PCA-Bench), visual question answering (A-OKVQA), multimodal understanding and reasoning (MMMU) and then prompt the LVLM to generate N responses $\hat{Y} = \{\hat{y}^1, ..., \hat{y}^N\}$.

During the decoding process of one response, we collect each output tokens' attention score distribution α in different layers and sum up for different type of input tokens. That is, for the i-th token, in the j-th layer, we compute $\alpha_{sys}^{i,j}, \alpha_{img}^{i,j}, \alpha_{ins}^{i,j}, \alpha_{out}^{i,j}$ to denote the total attention score current token attends to the system prompt, image tokens, user instruction and output tokens. We have:

$$\alpha_{sys}^{i,j} + \alpha_{img}^{i,j} + \alpha_{ins}^{i,j} + \alpha_{out}^{i,j} = 1 \tag{2}$$

We compute the total attention allocation λ to denote the total attention score one type of tokens received in one layer. For example, the total attention of system prompt in layer j is:

$$\lambda_{sys}^{j} = \sum_{i=1}^{n} \alpha_{sys}^{i,j} \tag{3}$$

where n is the number of tokens in the response. Final attention allocation is averaged over all attention heads in the N image-text pairs we sampled.

Next, we define metric **attention efficiency** ϵ to denote the average attention score per type's token received in one layer during the decoding process of one response. For example, the attention efficiency of image tokens in layer j is:

$$\epsilon_{img}^{j} = \frac{\sum_{i=1}^{n} \alpha_{img}^{i,j}}{|img|} \tag{4}$$

where $|img|$ is the number of image tokens, n is the number of tokens in the response. Final attention efficiency is averaged over all attention heads in the N image-text pairs we sampled.

In our experiment, N is set to 1000 and we use LLaVA1.5-7B as the LVLM. We follow the same generation configuration as the original paper [27].

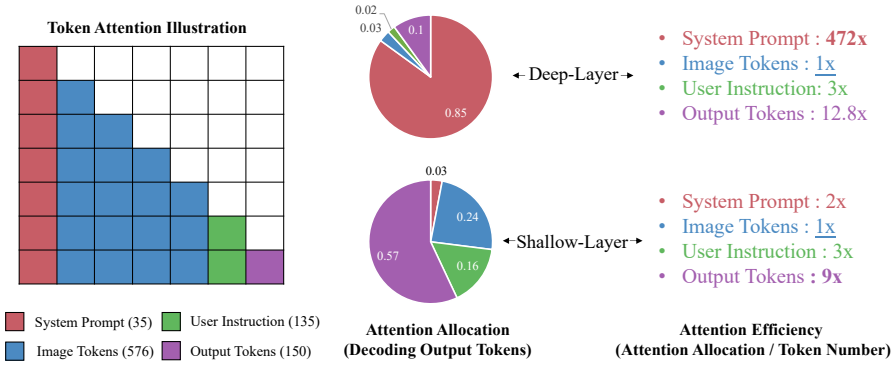

Fig. 3: Illustration of inefficient visual attention phenomena. The left part shows the relative position and average number of different type of input tokens, tokens could only attend to preceding tokens in the self-attention module. In average, image tokens take up most of the input tokens (64%). The middle and right part show the average attention allocation λ and attention efficiency ϵ in shallow and deep layers. Image tokens receive far less attention relative to their number in the deep layers.

3.3 Results

We have two major findings in the attention pattern statistics regrading attention allocation λ and attention efficiency ϵ for different type of input tokens. We define the first 2 layers as shallow layer and the rest 30 layers as deep layers.

1. Both attention allocation and attention efficiency show different degree of imbalance, which is related to the layer depth. The average attention allocation and efficiency in different layer is shown in Figure 3. In shallow layer the attention allocation is relatively more balanced than in deep layers. In shallow layer, the output tokens tends to attend to the previous output tokens while in deep layers, they tend to attend to the system prompt.
2. Image tokens have the **lowest** attention efficiency in both shallow and deep layers. System prompt is of extremely high attention efficiency in deep layers, which is **472** times that of image tokens, taking up 85% total attention scores.

3.4 Insights

The statistics reveal a surprising trend in the decoding process of LVLMs: despite accounting for the majority of tokens in the input, image tokens receive significantly less attention. Conversely, system prompts, which provides the minimal semantic information, attract the most of the attention scores. To delve deeper into this phenomenon, we analyze the attention maps of the first, middle, and last layers during during the decoding process of a model response as shown in Figure 4. The attention maps for all layers are provided in figure-7 of the supplement material.

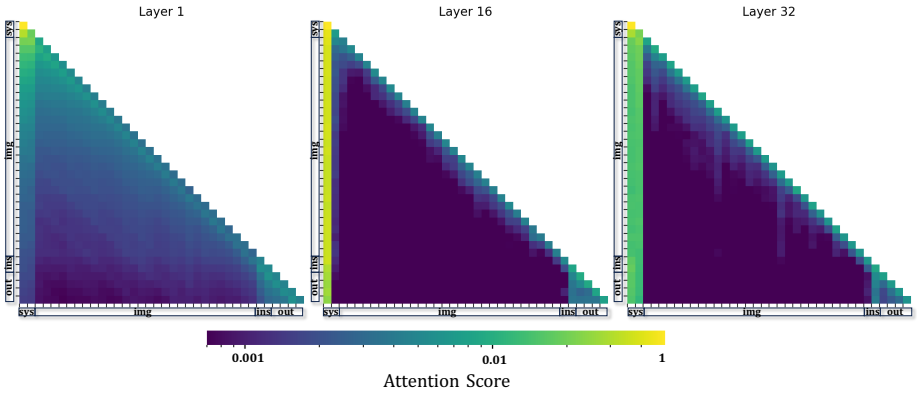

Fig. 4: The attention maps during the decoding process of one model response for LLaVA1.5-7B. We can see that in the bottom layer, attention distributes relatively smooth across different type of tokens. In the the deep layers, above from local attention, the attention scores are aggregated to system prompt, instruction and output tokens and attention over image tokens is rather sparse.

From the attention visualization results, we can see that in shallow layer, the attention scores distribute more smoothly across different tokens. While in deep layer, there are vertical strong lines (in the system prompt) that takes up most of attention scores. The existence of vertical strong line shows that there are some input tokens that consistently received high attention during the whole decoding process. This also explains the highly imbalanced attention efficiencies in our statistics: A small portion of anchor tokens aggregate the information from all input tokens and the model much favors to attend to those anchor tokens in deep layers. Our findings also align with the information flow of Large Language Model found in [40].

4 FastV

With insights from the validated phenomena and explanation, we propose FastV as a solution to reduce the inference costs of LVLMs without sacrificing the performance.

4.1 Dynamically Prune Vision Tokens

Figure 5 illustrates the general idea of FastV. The key is the image token re-rank and filtering module. It consists of one ranking function f_ϕ and two parameters: filtering layer K and filtering ratio $R\%$. At layer K of the LVLM, the ranking function f takes a sequence of input tokens and rank them by certain importance criteria ϕ. The last $R\%$ tokens after ranking would be pruned out in successive layers. We simply compute the average attention-score one token received from

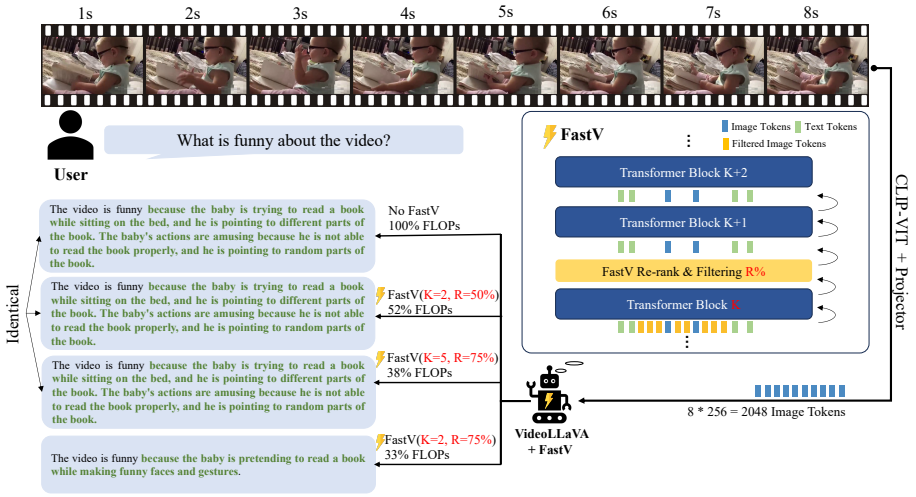

Fig. 5: Illustration of FastV. For image or video input (multiple image frames), they are first transformed to visual tokens with a pretrained image encoder like CLIP-VIT and then processed by the LLM decoder. FastV dynamically prunes $R\%$ image tokens after layer K in the forward process of input tokens. We can tell from the output that FastV does not influence the correctness while reducing significant FLOPs. The correct facts in the outputs are marked **green**. The first three outputs are completely identical.

all other tokens as the criteria ϕ_{attn} in our experiment. In extreme condition, K could be also set to 0, that image tokens are pruned before sending to the language model, we use random ranking as the criteria ϕ_{rand} where image tokens are randomly dropped.

FastV is plug-and-play to different token-based LVLMs for various vision language tasks without the need of training the model. We take video understanding tasks with VideoLLaVA [22] as example as shown in Figure 5.

4.2 Computing Cost Estimation

We consider the computation of multi-head attention (MHA) and feed-forward network (FFN) module in the FLOPs estimation. For one transformer layer, assume n is the token number, d is the hidden state size, m is the intermediate size of FFN, the total FLOPs can be estimated by $4nd^2 + 2n^2d + 2ndm$. For the whole model, assume FastV prunes tokens from n to $\hat{n} = (1 - R\%) \cdot n$ after layer K and there are T layers at all. The theoretical FLOPs reduction ratio related to image tokens is computed as:

$$1 - \frac{K \times (4nd^2 + 2n^2d + 2ndm) + (T - K) \times (4\hat{n}d^2 + 2\hat{n}^2d + 2\hat{n}dm)}{T \times (4nd^2 + 2n^2d + 2ndm)} \tag{5}$$

Table 1: Performance/Computation Balance of FastV under different configurations (K for filtering layer, R for filtering ratio). Highest score for each model is in red while the second highest is in blue.

Model	FastV Settings				Nocaps	Flickr30k	A-OKVQA	MMMU	Avg
	K	R	Flops(B)	Flops Ratio	*CIDEr*	*CIDEr*	*Accuracy*	*Accuracy*	
LLaVA-1.5-7B	Baseline		99.3	100%	99.8	67.9	76.7	34.8	**69.8**
	2	90%	19.9	20%	72.1	43.7	70.1	**35**	55.2
	2	75%	32.8	33%	94.6	63.6	75.5	34.8	67.1
	2	50%	54.6	55%	99.7	67.5	**77**	34.4	69.7
	3	90%	22.8	23%	87.2	55.8	71.9	34.8	62.4
	3	75%	34.8	35%	98	65	74.7	34.1	68.0
	3	50%	56.6	57%	99.7	**68.3**	76.7	34.3	**69.8**
	5	90%	27.8	28%	88.6	59.3	70.6	33.9	63.1
	5	75%	39.7	40%	98.5	66.3	74.8	34.3	68.5
	5	50%	59.6	60%	99.2	67.9	76.8	34.3	69.6
	0	90%	18.9	19%	7	53.2	66.8	34.7	40.4
	0	75%	28.8	29%	27.2	61.4	72.8	35.1	49.1
	0	50%	51.6	52%	**100.9**	65.5	75.3	34.3	69.0
LLaVA-1.5-13B	Baseline		154.6	100%	102.8	73	**82**	36.4	**73.6**
	2	90%	29.7	19%	87.9	62	75	36.3	65.3
	2	75%	50.2	32%	100.5	72.5	80.9	**38.1**	73.0
	2	50%	84.6	55%	**103.1**	73.4	81	36.7	**73.6**
	3	90%	33.0	21%	90.2	63.6	75.2	34.9	66.0
	3	75%	52.9	34%	100.9	72.1	79.5	36.4	72.2
	3	50%	86.4	56%	102.7	73.4	81.3	36.4	73.5
	5	90%	39.6	26%	93.5	67.4	75.8	35.4	68.0
	5	75%	58.4	38%	101.4	72.5	80	36.2	72.5
	5	50%	90.1	58%	102.5	**73.5**	81.2	36.6	73.5
QwenVL-Chat-7B	Baseline		71.9	100%	**94.9**	**72.5**	**75.6**	**35.8**	**69.7**
	2	90%	15.8	22%	81.9	61.5	68.5	35.3	61.7
	2	75%	24.4	34%	90.5	67.0	75.1	35.3	67.0
	2	50%	39.5	55%	94.4	71.4	75.3	35.6	69.2

We plot a 3D graph to show how the FLOPs reduction ratio changes with FastV's parameter K and R in Figure 8 from the supplement material.

4.3 Comparison: Training With Less Visual Tokens

FastV achieves computation reduction through eliminating redundant visual tokens during inference stage. An alternative method to reduce visual tokens is directly training with less visual tokens. This could be simply done by conducting pooling on the output of visual encoder during LVLM's training process. We compare FastV and this method in our ablation studies (sec. 5.4).

5 Experiments and Results

5.1 Evaluation Tasks

We conduct a wide range of evaluation including image captioning, VQA, multimodal reasoning, video QA and fine-grained benchmarks like MME [10] to examine the influence of FastV on the performance of LVLMs. We use greedy search for all experiments and provide details for each task in section A in the supplement material.

Table 2: Experiments with more models and benchmarks.

Methods	AI2Diagram ↑	SciQA-IMG ↑	SeedBench ↑	MMVet↑	MME ↑
LLaVA-1.5-13B	**59.45**	72.99	**68.23**	30.55	1827.75
+ FastV (K=2,R=50%)	58.96	**73.23**	68.03	**31.25**	**1849.68**
InstructBLIP-Vicuna-13B	**45.46**	61.15	**52.11**	**24.19**	**1143.5**
+ FastV (K=2,R=50%)	43.12	61.23	50.41	22.15	1129.8
+ FastV (K=5,R=50%)	44.39	**62.33**	51.69	23.51	1140.5

Table 3: Fine-grained results on MME benchmark.

Methods	Exist.	Count	Position	Color	OCR	Poster	Celeb.	Scene	Landmark	Art.	Comm.	Num.	Text.	Code.	Total
LLaVA-1.5-13B	**185.00**	**155.00**	**133.33**	170.00	125.00	**160.72**	152.54	161.25	**170.50**	**118.50**	**128.41**	**42.50**	77.50	47.50	1827.75
+ FastV (K=2,R=50%)	**185.00**	**155.00**	**133.33**	**175.00**	**132.50**	159.77	**153.15**	**161.75**	168.25	117.00	126.43	**42.50**	**82.50**	**57.50**	**1849.68**

5.2 Model Settings

We test FastV with various open source models. For image understanding tasks, we conduct experiments on LLaVA1.5-7B, 13B [25], and Qwen-VL [2]. When it comes to video understanding tasks, our baseline model is VideoLLaVA [22]. We adopt the settings as reported in their paper for the baseline models.

5.3 Main Results

Image Understanding. The performance on tasks under different FastV settings are shown in Table 1 (Nocaps, Flickr30k, A-OKVQA, MMMU) and Table 5 (PCA-Bench, OCR-VQA). The result of latency test is shown in Table 4.

In Table 1, we present the performance trend with FLOPs ratio ranging from 19% to 100% by FastV, for different type and size of models. We also plot the relation between FLOPs Reduction ratio (1-FLOPs Ratio) and average performance in Figure 1. The results indicate that FastV (K=2, R=50%) could achieve about 45% FLOPs reduction for different LVLMs without sacrificing the performance. The FLOPs-Performance trade-off is is also highly adjustable by lowering K and increasing R if we want to pursue an ultimate speed up. As shown in the latency test (Table 4), an 13B model with FastV could inference as fast as a 7B model with superior performance for A-OKVQA.

In PCA-Bench and OCR-VQA, (Table 5), which runs finegrained analysis on perception, cognition, action and OCR abilities, we find that FastV (K=2, R=50%) could maintain the sub-scores while significantly decreasing the FLOPs.

Video Understanding. The results of FastV on different video question answering tasks in shown in table 6 (TGIF, MSVD, MSRVTT). To our surprise, we find FastV could generally improves the Video-QA tasks performance while saving 40%+ computations especially for the TGIF task. We think the main reason is that the redundancy information problem is more severe for video understanding as multiple images from the video are transformed to tokens when sending to the LLM. For example, an image costs 576 tokens in LLaVA1.5 model,

Table 4: Real inference budget comparison between FastV and vanilla decoding. To get rid of the influence of output sequence length on decoding time, we report the result on A-OKVQA dataset where the model only needs to output an option. With FastV, an 13B model could inference as fast as a 7B model while maintaining its superior performance. The latency experiments are conducted on single A40 GPU.

Model	Total-Time	GPU-Memory	Score	Latency/Example
LLaVA-1.5-7B	6:34	19G	76.7	0.344s
w/ FastV (K=0, R=50%)	4:23	16G	75.3	0.230s
LLaVA-1.5-13B	10:17	38G	82.0	0.539s
w/ FastV (K=0, R=50%)	6:30	30G	80.5	0.341s

Table 5: Finegrained Results on PCA-Bench and OCR-VQA. P, C, and A each denotes Perception, Cognition and Action score. G-PCA denotes Genuine PCA score where the model must make correct perception, cognition and action for one test example to gain 1 score. The scores are averaged among all three domains including Auto-Driving, Domestic Robot and Open-World Game.

Model	FLOPs	PCA-Bench Open Test				PCA-Bench Closed Test				OCRVQA
		P	C	A	G-PCA	P	C	A	G-PCA	Rouge-L
LLaVA-1.5-7B	99.3B	0.493	0.353	0.433	0.263	0.513	0.387	0.450	0.277	0.51
LLaVA-1.5-13B	154.6B	0.530	**0.460**	0.503	0.333	0.563	**0.550**	0.573	0.353	**0.55**
w/ FastV (K=0, R=50%)	78.9B	0.490	0.395	0.443	0.292	0.519	0.450	0.512	0.283	0.49
w/ FastV (K=2, R=50%)	84.6B	**0.533**	0.423	**0.513**	**0.340**	**0.581**	0.545	**0.580**	**0.368**	**0.55**
w/ FastV (K=2, R=75%)	**50.2B**	0.513	0.417	0.483	0.320	0.523	0.510	0.533	0.323	0.54

while a video costs 2048 tokens in Video-LLaVA. As shown in the case from Figure 5, setting suitable FastV parameters could lead to much FLOPs reduction for Video-LLaVA while the outputs are nearly identical.

Fine-grained Benchmarks and More Models We conduct additional experiments with InstructBLIP and also with more fine-grained LVLM benchmarks such as SciQA-IMG [29], SeedBench [16], MMVet [45], and MME [10], together with benchmarks requiring more visual processing such as AI2Diagram. The results and fine-grained scores of MME are shown in Table 2 and Table 3. FastV works well on different LVLM benchmarks with competitive performance. We find that InstructBLIP shows slightly more performance degradation than LLaVA with same FastV config. The gap soon closes when we just set K to 5. We think it's because Q-Former initially reduces image tokens, resulting in direct information loss. Consequently, it requires adjusting the FastV parameters to avoid too much information loss.

5.4 Ablation Studies

Balance between Cost and Performance. We conduct an ablation experiment on how the parameters (K and R) influence the acceleration and downstream task's performance. We select OCR-VQA as the task, which necessitates a through understanding of the image. The result is shown in Figure 6. **When**

Table 6: GPT-Evaluation Results on Video Question Answering Tasks.

Model	TGIF		MSVD		MSRVTT		Avg	
	Acc	Score	Acc	Score	Acc	Score	Acc	Score
Video-LLaVA (Flops=100%)	0.18	2.5	0.70	3.9	0.56	**3.5**	0.48	3.3
w/ FastV (K=2, R=50%, Flops=52.3%)	**0.21**	**2.6**	**0.71**	3.9	0.55	**3.5**	**0.49**	3.3
w/ FastV (K=5, R=50%, Flops=57.1%)	0.20	**2.6**	**0.71**	**4.0**	**0.57**	**3.5**	**0.49**	**3.4**

Table 7: Ablation studies results. Scores labelled as "Failed" denotes the model could not follow instructions to generates valid results for evaluation.

Model	Nocaps	Flickr30k	A-OKVQA	MMMU
LLaVA1.5-7B (Retrained)	100.3	70.2	78.5	34.5
(a) w/ Train with 50% image tokens	98.5	68.5	76.8	33.5
(b) w/ FastV (K=2, R=50%)	**100.1**	**70**	**78.4**	**34.6**
(c) w/ FastV (K=2, R=50%, Random)	99.5	68.3	78.2	34.2
(d) w/ FastV (system prompt)	89.2	64.3	69.2	33.8
(e) w/ FastV (prune first half system prompt)	17.5	27.8	Failed	Failed
(f) w/ FastV (instruction)	77.3	50.1	56.5	29.5
(g) w/ StreamingLLM [41]	13.2	21.4	Failed	Failed

K is small, lowering R would improve the performance with a smaller FLOPs reduction ratio. In contrast, when K is large, adjusting R has minimal impact on the overall performance. This observation further proves that in deep layers, there is high redundancy in image tokens.

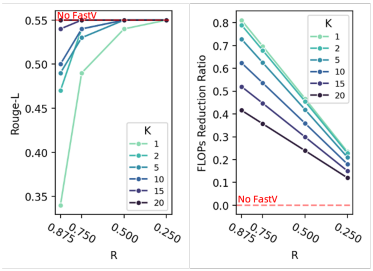

Fig. 6: Ablation study on filtering layer K and filtering ratio R in FastV. Experiments are conducted with LLaVA1.5-13B on OCR-VQA task. When K is small, lowering R would improve the performance with a smaller FLOPs reduction ratio. In contrast, when K is large, changing R has minimal impact on the overall performance.

Training with Less Tokens. FastV reduces computational requirements (FLOPs) by pruning tokens during the inference stage. An alternative approach for token reduction involves training the LVLM at a lower resolution. To facilitate a fair comparison, we retrained two LLaVA1.5-7B models, adhering to the orig-

inal pretraining and supervised finetuning protocols. The sole modification in the second model's training process was the incorporation of an average pooling layer (with a stride of 2) following the Clip encoder, leading to a 50% reduction in image tokens during training. A comparison between lines (a) and (b) in Table 7 reveals that reducing the input resolution directly during training results in diminished performance. Conversely, FastV manages to decrease the number of image tokens without compromising performance, showcasing its efficiency in balancing computational savings with model efficacy.

Pruning Token Strategy. FastV strategically reduces the number of image tokens during the inference phase of LVLMs, motivated by our observation that image tokens exhibit the lowest attention efficiency relative to other types of input tokens. In experiments detailed in lines (d) and (f) of the study, we specifically pruned tokens that were not related to images, such as system prompts and instruction tokens. This selective pruning resulted in significant performance declines, even when only a minimal number of non-image tokens were removed. We also compare randomly drop visual tokens instead of dropping by attention rank, as shown in line (c). It resulted in declined results compared with origin FastV (b). These findings underscore the distinct roles that visual and textual tokens play within LVLMs. It highlights FastV's effectiveness in precisely targeting image tokens for reduction, thereby optimizing performance without compromising the model's overall functionality.

In our previous observation about attention efficiency, we find out that the system prompt takes up of most attention even if they carry the least semantic information in the context. We conduct another experiment by directly prune the first half tokens of the system prompt. Comparing line (d) and (e), we can find that the head tokens in the system prompt have dominant effect on the model performance. Our findings also align with StreamingLLM [41] where they find that the first 4 tokens in LLM play the most important role during inference. However, direcly applying the same sparse attention pattern as StreamingLLM would lead to a substantial degradation in LVLM's performance as shown in line (g) of Table 7. This suggests a fundamental difference in how image tokens, as opposed to text tokens, contribute to the information processing within LLMs.

6 Conclusion

In this paper, we propose FastV, a plug-and-play inference cost optimization method for Large Vision-Language Models. Our insight for FastV arises from our observation that the attention computation over visual tokens is of extreme inefficiency in the deep layers of popular LVLMs though they take up a large portion of input tokens. FastV prunes out the unnecessary visual tokens according to the attention score ranking, which results in significant inference cost reduction without sacrificing performance.

Acknowledgments

We thank all reviewers for their valuable feedback. This work is supported by the National Science Foundation of China under Grant No.61936012 and 61876004.

References

1. Agrawal, H., Anderson, P., Desai, K., Wang, Y., Chen, X., Jain, R., Johnson, M., Batra, D., Parikh, D., Lee, S.: nocaps: novel object captioning at scale. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 8947–8956 (2019) 3, 20
2. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A frontier large vision-language model with versatile abilities. ArXiv preprint **abs/2308.12966** (2023) 1, 2, 5, 11
3. Bavishi, R., Elsen, E., Hawthorne, C., Nye, M., Odena, A., Somani, A., Taşırlar, S.: Introducing our multimodal models (2023), https://www.adept.ai/blog/fuyu-8b 4
4. Cao, Q., Paranjape, B., Hajishirzi, H.: Pumer: Pruning and merging tokens for efficient vision language models (2023), https://arxiv.org/abs/2305.17530 4
5. Chen, L., Zhang, Y., Ren, S., Zhao, H., Cai, Z., Wang, Y., Wang, P., Liu, T., Chang, B.: Towards end-to-end embodied decision making via multi-modal large language model: Explorations with gpt4-vision and beyond. ArXiv (2023) 3
6. Chen, L., Zhang, Y., Ren, S., Zhao, H., Cai, Z., Wang, Y., Wang, P., Meng, X., Liu, T., Chang, B.: Pca-bench: Evaluating multimodal large language models in perception-cognition-action chain (2024) 1, 3, 20
7. Dao, T.: Flashattention-2: Faster attention with better parallelism and work partitioning (2023) 4
8. Dao, T., Fu, D.Y., Ermon, S., Rudra, A., Ré, C.: Flashattention: Fast and memory-efficient exact attention with io-awareness (2022) 4
9. Driess, D., Xia, F., Sajjadi, M.S.M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., Florence, P.: Palm-e: An embodied multimodal language model. vol. abs/2303.03378 (2023) 1
10. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Qiu, Z., Lin, W., Yang, J., Zheng, X., et al.: Mme: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint arXiv:2306.13394 (2023) 3, 10, 12, 20
11. Ge, S., Zhang, Y., Liu, L., Zhang, M., Han, J., Gao, J.: Model tells you what to discard: Adaptive kv cache compression for llms (2024) 4
12. Jang, Y., Song, Y., Yu, Y., Kim, Y., Kim, G.: Tgif-qa: Toward spatio-temporal reasoning in visual question answering (2017) 3, 20
13. Kondratyuk, D., Yu, L., Gu, X., Lezama, J., Huang, J., Hornung, R., Adam, H., Akbari, H., Alon, Y., Birodkar, V., Cheng, Y., Chiu, M.C., Dillon, J., Essa, I., Gupta, A., Hahn, M., Hauth, A., Hendon, D., Martinez, A., Minnen, D., Ross, D., Schindler, G., Sirotenko, M., Sohn, K., Somandepalli, K., Wang, H., Yan, J., Yang, M.H., Yang, X., Seybold, B., Jiang, L.: Videopoet: A large language model for zero-shot video generation (2023) 4
14. Kong, Z., Dong, P., Ma, X., Meng, X., Sun, M., Niu, W., Shen, X., Yuan, G., Ren, B., Qin, M., Tang, H., Wang, Y.: Spvit: Enabling faster vision transformers via soft token pruning (2022), https://arxiv.org/abs/2112.13890 4

15. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C.H., Gonzalez, J.E., Zhang, H., Stoica, I.: Efficient memory management for large language model serving with pagedattention (2023) 4, 21
16. Li, B., Wang, R., Wang, G., Ge, Y., Ge, Y., Shan, Y.: Seed-bench: Benchmarking multimodal llms with generative comprehension (2023), https://arxiv.org/abs/2307.16125 3, 12, 20
17. Li, J., Pan, K., Ge, Z., Gao, M., Zhang, H., Ji, W., Zhang, W., Chua, T.S., Tang, S., Zhuang, Y.: Empowering vision-language models to follow interleaved vision-language instructions. arXiv preprint arXiv:2308.04152 (2023) 4
18. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. ArXiv preprint **abs/2301.12597** (2023) 4, 5
19. Li, Y., Wang, C., Jia, J.: Llama-vid: An image is worth 2 tokens in large language models (2023) 4
20. Li, Z., Yang, B., Liu, Q., Ma, Z., Zhang, S., Yang, J., Sun, Y., Liu, Y., Bai, X.: Monkey: Image resolution and text label are important things for large multi-modal models. arXiv preprint arXiv:2311.06607 (2023) 3
21. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Not all patches are what you need: Expediting vision transformers via token reorganizations (2022), https://arxiv.org/abs/2202.07800 4
22. Lin, B., Zhu, B., Ye, Y., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning united visual representation by alignment before projection. arXiv preprint arXiv:2311.10122 (2023) 9, 11
23. Liu, H., Yan, W., Zaharia, M., Abbeel, P.: World model on million-length video and language with ringattention (2024) 1, 4
24. Liu, H., Zaharia, M., Abbeel, P.: Ring attention with blockwise transformers for near-infinite context (2023) 4
25. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning (2023) 4, 5, 11
26. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (January 2024), https://llava-vl.github.io/blog/2024-01-30-llava-next/ 4
27. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. ArXiv preprint **abs/2304.08485** (2023) 1, 2, 3, 4, 6
28. Lu, J., Clark, C., Lee, S., Zhang, Z., Khosla, S., Marten, R., Hoiem, D., Kembhavi, A.: Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action (2023) 4
29. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: Multimodal reasoning via thought chains for science question answering. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 2507–2521. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/11332b6b6cf4485b84afadb1352d3a9a-Paper-Conference.pdf 12, 20
30. Maaz, M., Rasheed, H., Khan, S., Khan, F.S.: Video-chatgpt: Towards detailed video understanding via large vision and language models. arXiv:2306.05424 (2023) 20
31. Mishra, A., Shekhar, S., Singh, A.K., Chakraborty, A.: Ocr-vqa: Visual question answering by reading text in images. In: 2019 international conference on document analysis and recognition (ICDAR). pp. 947–952. IEEE (2019) 3, 20

32. OpenAI: Gpt-4v(ision) system card (2023) 2
33. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE international conference on computer vision. pp. 2641–2649 (2015) 3, 20
34. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763 (2021) 5
35. Schwenk, D., Khandelwal, A., Clark, C., Marino, K., Mottaghi, R.: A-okvqa: A benchmark for visual question answering using world knowledge. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII. pp. 146–162. Springer (2022) 3, 20
36. Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023) 1, 4
37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 5998–6008 (2017) 5
38. Vedantam, R., Zitnick, C.L., Parikh, D.: Cider: Consensus-based image description evaluation (2015) 20
39. Wang, J., Xu, H., Ye, J., Yan, M., Shen, W., Zhang, J., Huang, F., Sang, J.: Mobile-agent: Autonomous multi-modal mobile device agent with visual perception (2024) 1
40. Wang, L., Li, L., Dai, D., Chen, D., Zhou, H., Meng, F., Zhou, J., Sun, X.: Label words are anchors: An information flow perspective for understanding in-context learning. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 9840–9855. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.emnlp-main.609, https://aclanthology.org/2023.emnlp-main.609 8
41. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. arXiv (2023) 4, 13, 14
42. Xiong, Y., Chen, H., Hao, T., Lin, Z., Han, J., Zhang, Y., Wang, G., Bao, Y., Ding, G.: Pyra: Parallel yielding re-activation for training-inference efficient task adaptation (2024), https://arxiv.org/abs/2403.09192 4
43. Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., Zhuang, Y.: Video question answering via gradually refined attention over appearance and motion. In: Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017. pp. 1645–1653 (2017) 3, 20
44. Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., Zhuang, Y.: Video question answering via gradually refined attention over appearance and motion. In: ACM Multimedia (2017) 3, 20
45. Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., Wang, L.: Mm-vet: Evaluating large multimodal models for integrated capabilities (2023), https://arxiv.org/abs/2308.02490 3, 12, 20

46. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., Wei, C., Yu, B., Yuan, R., Sun, R., Yin, M., Zheng, B., Yang, Z., Liu, Y., Huang, W., Sun, H., Su, Y., Chen, W.: Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. arXiv preprint arXiv:2311.16502 (2023) 3
47. Zhao, H., Cai, Z., Si, S., Ma, X., An, K., Chen, L., Liu, Z., Wang, S., Han, W., Chang, B.: Mmicl: Empowering vision-language model with multi-modal in-context learning. ArXiv preprint **abs/2309.07915** (2023) 2
48. Zheng, B., Gou, B., Kil, J., Sun, H., Su, Y.: Gpt-4v(ision) is a generalist web agent, if grounded (2024) 1
49. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. ArXiv preprint **abs/2304.10592** (2023) 2, 5