Accelerating Multimodal Large Language Models by Searching Optimal Vision Token Reduction

Supplementary Material

This document supplements the main paper as follows.

- Sec. 6.1 provides the detailed results of G-Search on 12 benchmarks.
- Sec. 6.2 applies our method on larger MLLMs, *e.g.*, LLaVA-1.5-13B, and InternVL2-26B. Our method consistently boosts the efficiency of MLLMs in various sizes.
- Sec. 6.3 provides the detailed results to show that our G-Search can further improve the efficiency of MLLMs on top of prompt-agnostic methods.
- Sec. 7.1 provides more metrics for efficiency evaluations, *e.g.*, Multiply Accumulate Operations (MACs), and per-token decoding time cost.
- Sec. 7.2 compares our method with the I/O aware approach FlashAttention2 [9, 10]. Our method runs faster and is able to trade off effectiveness and efficiency.
- Sec. 8 evaluates our method on video benchmarks. The reduction strategy searched by our method on image understanding generalizes to video understanding.
- Sec. 9.1 shows how to set the budget for P-Sigmoid with theoretical analysis.
- Sec. 9.2 provides the values of k found by our P-Sigmoid. The value varies by MLLMs.
- Sec. 9.3 illustrates the correlations of MLLM layers in terms of vision tokens sorted by attention scores. The main finding of this paper holds for various MLLMs.
- Sec. 9.4 evaluates G-Search on the long context benchmark, MM-NIAH [40].
- Sec. 9.5 shows the efficiency of G-Search with various token length.
- Sec. 9.6 talks about limitations of this paper and provides potential solutions.

6. Detailed results on 12 benchmarks

6.1. G-Search and existing prompt-aware methods

Table 6 provides the concrete results on the 12 image understanding benchmarks for the proposed G-Search and several prompt-aware vision token reduction methods, *i.e.*, VTW [26], PDrop [44], and FastV [5], on top of various MLLMs. The results are complementary to Table 1 of the main paper. Our method consistently reduce computational cost without significant performance drops, while other methods may fail in preserving the performance.

6.2. G-Search on larger MLLMs.

Besides MLLMs explored in the main paper, we apply G-Search on top of larger MLLMs, *e.g.*, LLaVA-1.5-13B and InternVL2-26B, and report the results in the last two blocks of Table 6. The performance of InternVL2-26B on MMMU is not reported because it runs out of GPU memories during inference probably due to too long contexts. Thus, we compute the average accuracy for InternVL2-26B on the rest 11 benchmarks. The results on larger MLLMs are in line with those on smaller MLLMs. That is, the proposed G-Search requires less computations without a significant performance drop, while other methods either require much more computations or suffer from performance drops. Such results indicate that our method is robust on various sizes of models and scales up well.

6.3. G-Search on top of prompt-agnostic methods

Table 7 provides the concrete results of applying the proposed G-Search on two prompt-agnostic vision token reduction methods, *i.e.*, TokenPacker [20], and DeCo [46]. The results are complementary to Table 2 of the main paper. As one can see, our G-Search significantly reduces the computational cost with slight variances in performance on all benchmarks, which clearly demonstrates that our method is flexible and robust in various cases.

7. Extra evaluations of computational costs

Not only can our methods reduce the number of vision tokens to accelerate the prefilling phrase of the inference, but also it reduces the KV-Cache to speed up the decoding phrase. In addition to the prefilling time cost in the main paper, we report more metrics for computational costs in this section, including MACs, and the per-token time cost at the decoding stage.

7.1. Comparison to existing reduction methods.

Table 8 provides more metrics to evaluate the efficiency for Scenario I. The proposed G-Search is compared with other completing reduction methods, *i.e.*, VTW [26], PDrop [44], and FastV [5]. The results are complementary to Table 1 of the main paper. As shown in Table 8, our method achieves good per-token decoding time cost, as well as other metrics like prefilling time cost.

Table 9 provides more efficiency metrics for Scenario II. As a complement to Table 3 of the main paper, we compare P-Sigmoid with the prior SOTA FastV (R=87.5%) on top of various MLLMs. The budget of P-Sigmoid is set similar as FastV. As we can see, compared to FastV, P-Sigmoid achieves much better performance with similar or slightly less memory cost, TFLOPs, MACs and time costs.

Base MLLM + Method			General VQA			Knowledge			OCR & Chart			Vision-Centric		
	TFLOPs ↓	Avg acc. ↑	MME	MMBench	GQA	MMMU	MathVista	AI2D	TextVQA	ChartQA	DocVQA	POPE	RealWorldQA	HallusionBench
LLaVA-1.5-7B	9.18	48.97	1743.2	65.8	62.8	37.6	21.6	55.6	46.4	17.7	29.1	86.1	55.0	47.6
+ VTW	5.19	44.32	1740.6	65.9	56.4	37.8	21.2	55.5	16.5	13.8	14.1	86.0	54.6	47.7
+ PDrop	4.95	48.70	1714.9	65.5	61.9	38.0	21.9	55.2	46.4	17.2	28.9	86.1	55.5	46.6
+ FastV (R=50%)	5.47	48.70	1760.8	64.8	61.8	37.7	21.8	55.6	46.2	17.9	27.8	84.7	56.2	47.1
+ G-Search (Ours)	**3.95**	**48.77**	1741.1	65.3	62.1	37.7	22.0	55.5	46.0	17.8	28.1	85.5	56.3	46.7
InternVL2-1B	4.62	59.85	1778.3	63.2	55.1	34.6	32.6	62.6	69.1	71.3	82.0	87.3	51.5	45.3
+ VTW	**3.73**	41.13	1767.5	62.9	42.5	33.8	20.9	60.7	11.7	12.2	11.7	83.9	46.4	43.7
+ PDrop	3.88	53.70	1764.3	62.0	53.4	34.7	29.2	59.5	59.3	51.0	52.4	86.7	49.9	43.3
+ FastV (R=50%)	3.91	54.85	1747.4	61.8	53.8	34.9	30.9	59.8	62.1	58.7	55.8	85.1	48.4	44.6
+ G-Search (Ours)	3.84	**59.19**	1750.7	62.5	55.0	34.8	32.5	62.1	68.3	69.4	79.0	87.0	51.9	45.2
InternVL2-2B	8.10	61.94	1821.7	72.5	59.9	34.7	33.8	72.5	72.0	75.0	87.2	88.4	34.0	48.3
+ VTW	**5.50**	37.84	1596.3	64.2	42.6	32.3	25.0	66.8	10.7	11.0	11.3	67.8	22.9	42.5
+ PDrop	5.93	58.98	1800.1	71.5	58.2	33.9	32.6	70.6	70.2	67.6	73.7	85.9	33.2	46.2
+ FastV (R=50%)	5.85	59.91	1774.2	71.8	58.7	34.1	33.1	71.6	70.7	69.6	75.7	88.0	32.9	49.3
+ G-Search (Ours)	5.64	**61.22**	1831.7	71.9	59.4	34.8	33.6	71.9	71.6	71.6	84.4	88.2	34.0	47.7
InternVL2-4B	13.97	68.16	2084.4	77.6	62.0	45.8	36.4	77.8	74.1	81.0	89.6	87.1	59.7	52.3
+ VTW	8.72	49.01	2027.9	76.4	51.1	45.8	26.7	77.3	14.7	14.8	15.2	85.2	56.5	52.0
+ PDrop	**8.35**	66.94	2084.5	77.4	61.4	46.3	35.5	77.0	72.7	77.4	82.2	87.0	60.9	50.9
+ FastV (R=50%)	9.01	66.19	2080.1	77.6	61.6	45.9	35.2	77.5	72.5	75.1	76.5	86.7	60.0	51.5
+ G-Search (Ours)	8.69	**67.65**	2082.3	77.4	61.8	46.0	36.4	77.5	73.9	80.2	86.9	87.0	59.5	50.8
InternVL2-8B	24.10	70.83	2205.3	81.8	62.7	48.6	36.9	82.4	76.6	82.6	91.9	86.7	65.5	55.5
+ VTW	13.71	52.51	2195.1	81.6	56.6	46.7	31.4	81.8	18.9	16.3	17.5	85.8	63.4	51.7
+ PDrop	13.13	69.19	2193.1	81.4	62.3	45.8	35.7	80.4	75.6	81.6	86.3	86.7	64.6	51.5
+ FastV (R=50%)	14.58	69.42	2214.2	81.2	62.0	48.1	37.3	81.1	75.6	80.2	83.4	86.5	64.6	53.8
+ G-Search (Ours)	**12.24**	**70.10**	2216.7	81.4	62.6	48.6	36.2	82.1	76.0	81.1	89.8	86.9	65.1	52.3
LLaVA-1.5-13B	17.44	49.77	1818.0	68.8	63.3	35.6	22.9	59.3	47.6	18.2	30.3	85.9	55.0	45.4
+ VTW	9.77	46.93	1828.1	68.7	60.1	35.4	22.6	59.3	33.7	15.5	15.1	86.0	55.7	45.6
+ PDrop	9.29	49.24	1810.4	68.4	63.0	36.1	23.3	59.1	47.6	18.3	23.5	86.0	55.3	45.6
+ FastV (R=50%)	10.18	**49.69**	1857.4	68.4	62.6	36.1	23.5	58.9	47.2	18.3	28.9	85.0	55.7	45.4
+ G-Search (Ours)	**6.45**	49.65	1835.5	68.2	62.4	36.6	23.7	58.9	47.6	18.5	29.0	85.8	54.4	45.1
InternVL2-26B	111.45	74.12	2272.8	81.8	65.2	-	37.8	83.1	82.0	84.7	90.5	88.0	67.7	53.4
+ VTW	84.94	67.12	2293.4	81.8	63.3	-	38.0	83.3	55.3	61.6	63.6	88.0	66.7	54.7
+ PDrop	83.30	73.12	2245.8	81.6	65.2	-	37.2	81.9	81.8	84.0	83.2	88.2	67.5	53.6
+ FastV (R=50%)	86.25	72.75	2244.6	81.3	64.8	-	37.5	82.1	81.0	83.0	82.5	87.4	67.2	53.4
+ G-Search (Ours)	**78.98**	**73.72**	2253.5	81.7	65.1	-	37.7	82.6	81.8	84.4	88.0	88.2	67.2	53.8

Table 6. Detailed results of prompt-aware methods on 12 benchmarks. InternVL2-26B on MMMU is not reported due to the out-of-memory issue on our platform. Average accuracy for InternVL2-26B is averaged on the other 11 benchmarks

7.2. Comparison to FlashAttention.

FlashAttention is a widely adopted I/O aware approach to accelerate transformer-based models like LLMs. Unlike token reduction methods, it lowers down the number of times to read and write memories instead of reducing FLOPs.

Evaluation on 12 benchmarks used in the main paper: We compare our method to the latest version of FlashAttention, *i.e*. FlashAttention2 [9], on top of LLaVA-1.5-7B and InternVL2-8B. We evaluate models with half-precision floating-point (specifically bfloat16 is used) because FlashAttention2 only supports this data format. Thus, the time cost of MLLMs in this section is lower than that

LLaVA-1.5-7B + Method	TFLOPs ↓	Avg acc. ↑	General VQA			Knowledge			OCR & Chart			Vision-Centric		
			MME	MMBench	GQA	MMMU	MathVista	AI2D	TextVQA	ChartQA	DocVQA	POPE	RealWorldQA	HallusionBench
TokenPacker	3.27	47.60	1726.1	64.5	61.7	37.1	21.9	55.0	41.7	16.8	23.2	86.1	53.5	48.0
+ G-Search (Ours)	2.12	47.68	1756.4	64.6	61.3	37.1	22.0	55.5	41.8	16.2	22.8	86.8	52.5	48.8
Δ	**-1.15**	**+0.08**	**+30.3**	**+0.1**	-0.4	0.0	**+0.1**	**+0.5**	**+0.1**	-0.6	-0.4	**+0.7**	-1.0	**+0.8**
DeCo	3.26	46.97	1714.1	64.3	61.4	37.2	21.4	54.8	40.1	15.8	22.4	84.9	53.3	46.9
+ G-Search (Ours)	2.16	46.71	1697.6	64.5	60.7	37.0	21.1	54.7	40.0	15.6	22.4	85.0	52.4	46.5
Δ	**-1.10**	-0.26	-16.5	**+0.2**	-0.7	-0.2	-0.3	-0.1	-0.1	-0.2	0.0	**+0.1**	-0.9	-0.4

Table 7. Detailed results for G-Search applied on top of prompt-agnostic methods.

Base MLLM + Method	Average accuracy↑	Memory cost↓	TFLOPs↓	TMACs↓	# Params (B)	Prefilling time cost↓	Decoding time cost↓
LLaVA-1.5-7B	48.97	1.000	9.18	4.59	6.76	0.625	0.181
+ VTW	44.32	0.500	5.19	2.60	6.76	0.385	0.134
+ PDrop	48.70	0.469	4.95	2.47	6.76	0.381	0.123
+ FastV (R=50%)	48.70	0.531	5.47	2.73	6.76	0.387	0.136
+ G-Search (Ours)	48.77	0.340	3.95	1.98	6.76	0.301	0.117
InternVL2-1B	59.85	1.000	4.62	2.31	0.94	0.384	0.123
+ VTW	41.13	0.500	3.73	1.86	0.94	0.331	0.114
+ PDrop	53.70	0.583	3.88	1.94	0.94	0.336	0.114
+ FastV (R=50%)	54.85	0.542	3.91	1.96	0.94	0.342	0.120
+ G-Search (Ours)	59.19	0.527	3.84	1.92	0.94	0.333	0.114
InternVL2-2B	61.94	1.000	8.10	4.05	2.21	0.598	0.172
+ VTW	37.84	0.500	5.50	2.75	2.21	0.439	0.138
+ PDrop	58.98	0.583	5.93	2.96	2.21	0.452	0.140
+ FastV (R=50%)	59.91	0.542	5.85	2.92	2.21	0.451	0.135
+ G-Search (Ours)	61.22	0.532	5.64	2.82	2.21	0.444	0.132
InternVL2-4B	68.16	1.000	13.97	6.98	4.15	0.969	0.256
+ VTW	49.01	0.500	8.72	4.36	4.15	0.649	0.189
+ PDrop	66.94	0.469	8.35	4.17	4.15	0.627	0.190
+ FastV (R=50%)	66.19	0.531	9.01	4.50	4.15	0.652	0.199
+ G-Search (Ours)	67.65	0.488	8.69	4.34	4.15	0.645	0.190
InternVL2-8B	70.83	1.000	24.10	12.05	8.08	1.518	0.388
+ VTW	52.51	0.500	13.71	6.85	8.08	0.927	0.251
+ PDrop	69.19	0.469	13.13	6.56	8.08	0.915	0.249
+ FastV (R=50%)	69.42	0.531	14.58	7.29	8.08	0.998	0.266
+ G-Search (Ours)	70.10	0.424	12.24	6.12	8.08	0.860	0.237

Table 8. More metrics of efficiency for Scenario I. We report the average accuracy calculated on 12 benchmarks and the per-token decoding time cost. Our method achieves good decoding time cost, as well as prefilling time cost.

in the main paper. As shown in Table 10, FlashAttention2 has slightly lower TFLOPs than the vanilla model. This is probably because FlashAttention2 dose not require the 4D attention masks to enable casual attentions. Although FlashAttention2 reduces time costs in both prefilling and decoding stages compared to the vanilla model, our method is faster without significant performance drops. Moreover, our method can be further enhanced with FlashAttention2 in the decoding stage. Besides, FlashAttention2 can only improve the efficiency for Scenario I where models are accelerated without performance drops. In contrast, our method works for both Scenario I and Scenario II. We believe Sce-

Base MLLM + Method	Average accuracy↑	Memory cost↓	TFLOPs↓	TMACs↓	# Params (B)	Prefilling time cost↓	Decoding time cost↓
LLaVA-1.5-7B	48.97	1.000	9.18	4.59	6.76	0.625	0.181
+ FastV	43.14	0.180	2.74	1.37	6.76	0.227	0.098
+ P-Sigmoid (Ours)	46.52	0.171	2.66	1.33	6.76	0.223	0.095
InternVL2-1B	59.85	1.000	4.62	2.31	0.94	0.384	0.123
+ FastV	43.16	0.198	3.43	1.71	0.94	0.308	0.108
+ P-Sigmoid (Ours)	47.83	0.188	3.38	1.69	0.94	0.304	0.106
InternVL2-2B	61.94	1.000	8.10	4.05	2.21	0.598	0.172
+ FastV	47.66	0.198	4.26	2.13	2.21	0.353	0.112
+ P-Sigmoid (Ours)	51.54	0.188	4.16	2.08	2.21	0.350	0.109
InternVL2-4B	68.16	1.000	13.97	6.98	4.15	0.969	0.256
+ FastV	54.83	0.180	5.48	2.74	4.15	0.441	0.143
+ P-Sigmoid (Ours)	61.19	0.171	5.38	2.69	4.15	0.436	0.141
InternVL2-8B	70.83	1.000	24.10	12.05	8.08	1.518	0.388
+ FastV	55.17	0.180	7.71	3.86	8.08	0.590	0.182
+ P-Sigmoid (Ours)	62.86	0.171	7.46	3.73	8.08	0.577	0.178

Table 9. More metrics of efficiency for Scenario II. We set R=87.5% for Fast V.

Base MLLM + Method	Average accuracy↑	TFLOPs↓	TMACs↓	# Params (B)	Prefilling time cost↓	Decoding time cost↓
LLaVA-1.5-7B	48.43	9.18	4.59	6.76	0.135	0.077
+ FlashAttention2	48.41	8.96	4.48	6.76	0.128	0.073
+ G-Search (Ours)	48.25	3.95	1.98	6.76	0.109	0.068
InternVL2-8B	70.39	24.10	12.05	8.08	0.278	0.101
+ FlashAttention2	70.34	23.67	11.84	8.08	0.212	0.085
+ G-Search (Ours)	70.07	12.24	6.12	8.08	0.163	0.074

Table 10. Comparison to FlashAttention2. Bflot16 is adopted to enable FlashAttention2. Our method achieves better efficiency with negligible performance drops. Furthermore, our method is able to trade off the efficiency and the performance for Scenario II.

nario II, where the performance is improved with given budgets, is in demand and important for edge applications, but it is ignored by current studies.

Evaluation with long contexts: FlashAttention can significantly accelerate MLLMs for long context scenarios. To leverage it, we slightly modify the implementation of our G-Search. We additionally calculate the attention scores between vision tokens and instruction tokens instead of getting the scores from attention layers. Since the instruction is short, the extra overhead is light. With the modified implementation, G-Search can be applied together with FlashAttention. Table 11 compares LongVA [51] our method and FlashAttention on VideoMME with around 32K tokens. G-Search further improves the efficiency upon FlashAttention by 2×.

8. Evaluation on video benchmarks

It is a common practice that MLLMs handle a video by sampling several images from the video and encoding each image into vision tokens. Such an approach leads to a large amount of redundant vision tokens, and should benefit from vision token reduction methods. In this section, we demonstrate our reduction method also accelerates MLLMs in video understanding. We evaluate our method on top of InternVL2-8B, which is trained with videos, on two popular video understanding benchmarks, *i.e.*, MVBench [19] and Video-MME [12]. The InternVL2 model will sample 8 images from the input video and encode them into more than two thousands of vision tokens. We leverage the reduction strategy from our G-Search, which finds the optimal keeping rates on image understanding data, to speed up the InternVL2 model.

Table 12 compares our method to FastV variants configured as R=50% and R=87.5%. We have the following interesting findings. First, we can reduce more computational costs for video understanding tasks than image understanding tasks. For example, P-Sigmoid requires less TFLOPs and runs faster than G-search. But they gain similar performance on the two video benchmarks. This is plausible

LongVA	Accuracy↑	Prefilling time↓	Decoding time↓	Max memory↓
+ FlashAttn	52.8	5.44 s	1.32 s	102 GB
+ FlashAttn & Ours	52.1	**2.35 s**	**0.61 s**	**63 GB**

Table 11. LongVA with around 32K tokens on VideoMME

Base MLLM + Method	Reduction Strategy	Effectiveness		Efficiency		
		MVBench↑	Video-MME↑	Memory cost↓	TFLOPs↓	Time cost↓
InternVL2-8B	None	64.67	52.41	1.0	38.71	2.522
+ FastV (R=50%)	Handcrafted	64.72	52.48	0.531	22.91	1.512
+ G-Search (Ours)	Automatic	64.65	52.52	0.424	19.13	1.340
Δ	-	-0.07	**+0.04**	**-0.107**	**-3.78**	**-0.172**
+ FastV (R=87.5%)	Handcrafted	63.88	51.63	0.180	11.74	0.891
+ P-Sigmoid (Ours)	Automatic	64.70	52.48	0.171	11.34	0.886
Δ	-	**+0.82**	**+0.85**	**-0.009**	**-0.40**	**-0.005**

Table 12. Evaluation two video benchmarks. Reduction methods are applied on top of InternVL2-8B that is trained with video data. G-Search achieves better efficiency compared to FastV with the default setting. P-Sigmod uses a similar budget as FastV (R=87.5%) and gains better performance.

because there more redundant vision tokens in videos than in images. Thus, more tokens can be removed without information loss. Second, for Scenario II, the performance gap between P-Sigmoid and FastV in video understanding is smaller than that in image understanding. Probably, this is caused by the fact that videos gain lots of redundant vision tokens. Non-optimal reduction strategy is likely to remove tokens without much information loss. Such results motivate a promising future work that explores how to remove highly redundant vision tokens.

9. More analysis & illustration

9.1. How to set budgets for P-Sigmoid

As mentioned in the main paper, to set the budgets of P-Sigmoid as the budgets of FastV [5], we first set the number of vision tokens in P-Sigmoid the same as that in FastV. Then, we slightly lower down the number of vision tokens to match the TFLOPs of P-Sigmoid and FastV. This is because the number of vision tokens is not exactly proportional to TFLOPs, as discussed below.

Per the discussion in FastV [5], the total FLOPs of i-th layer of a LLM is $C_i = 4n_id^2 + 2n_i^2d + 2n_idm$ where n_i is the number of tokens at this layer, d is the hidden state size of the multi-head attention, and m is the intermediate size of the feed-forward network. Thus, for a LLM with L layers, the total FLOPs C can be written as,

$$C = \sum_i^L (4n_id^2 + 2n_i^2d + 2n_idm) \tag{4}$$

$$= (4d^2 + 2dm)N + \sum_i^L n_i^2 \tag{5}$$

where $N = \sum_i n_i$ refers to the number of total tokens from all layers. The term $\sum_i^L n_i^2$ reaches the minimal when $\forall n_i = N/L$, which is the case of FastV. We provide a brief proof for the above statement below. According to Cauchy–Schwarz inequality,

$$(\sum_i^n x_iy_i)^2 \leq (\sum_i^n x_i^2)(\sum_i^n y_i^2). \tag{6}$$

When the inequality becomes an equality if and only if $\forall i, \forall j, x_i/y_i = x_j/y_j$. We set $y_i = 1$, $x_i = n_i$ and $n = L$, and we have

$$L(\sum_i^L n_i^2) \geq (\sum_i^L n_i)^2 = N^2. \tag{7}$$

When $\forall n_i = N/L$, $(\sum_i^L n_i^2)$ reaches the minimal N^2/L.

Since FastV reaches the minimal FLOPs for a given N, any other reduction strategies always have more FLOPs for the same N. Therefore, to match the FLOPs of FastV and our method, we have to reduce the number of total tokens N for our method.

9.2. Values of k for different MLLMs

Fig. 7 illustrates values of k from P-Sigmoid for Table 9. As shown, on top of LLaVA-1.5-7B and InternVL-8B, our

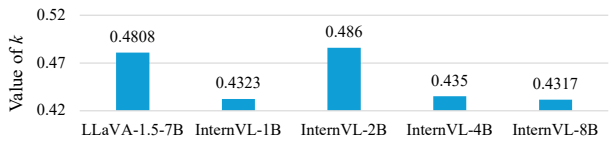


Figure 7. Values of k for different MLLMs from P-Sigmoid.

	1K	2K	4K	8K	16K	32K
InternVL-8B	70.7	58.1	48.6	46.0	37.7	24.2
+ Ours	70.9	58.3	49.0	46.4	**39.4**	**29.2**

Table 13. G-Search with various context lengths on MM-NIAH.

method has the same memory cost but outputs different k. On top of InternVL-1B and InternVL-2B, P-Sigmoid again outputs different k with the same memory cost. Those results indicate that different MLLMs should have different parameters and reduction strategies, which further explains why P-Sigmoid with automatic search can outperform other methods regardless of MLLMs.

9.3. Correlations between layers

In Sec. 1 of the main paper, we demonstrate our main finding by analyzing LLaVA-1.5-7B. As shown in Fig. 8a, we sort the vision tokens of each layer of LLaVA-1.5-7B based on their attention scores to instruction tokens and calculate the Kendall's Tau correlation coefficient [17] between the current layer and the next layer. We regard the relative importance of one vision token as its ranking. Then, the finding is that the relative importance of each vision token remains similar in each layer of MLLMs after the first layer. In this section, we provide the same visualization by analyzing InternVL2-1B/2B/4B/8B models. As shown in Fig. 8b, we get similar observations as Fig. 8a. As a result, our finding is a general case for different MLLMs.

In addition to correlation between consecutive layers, Fig. 9 visualizes the correlation coefficients between every two layers of the LLMs within different MLLMs. As shown, for all MLLMs, the coefficients are high in almost all regions except the first row and the first column. The result further enhance our finding about the relative importance of vision tokens.

9.4. Evaluation on long context benchmarks

We evaluated our G-Search on top of InternVL2-8B on MM-NIAH [40] in Table 13. As shown, with our method, the performance remains the same for short contexts ($\leq$ 8K), and improves for longer contexts. This is probably because our method removes irrelevant tokens and focuses on informative tokens. It is more likely to find the "needle" in a haystack with a few relevant tokens.

InternVL2-8B	Prefill (s)↓		Decode (s)↓		Max mem.↓	
w/ G-Search?	No	Yes	No	Yes	No	Yes
4K tokens	0.60	**0.36**	0.20	**0.15**	56 GB	**50 GB**
8K tokens	1.17	**0.62**	0.33	**0.21**	62 GB	**52 GB**
16K tokens	2.65	**1.13**	0.68	**0.32**	77 GB	**56 GB**

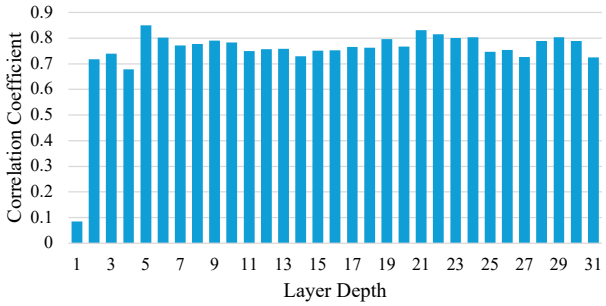
Table 14. Efficiency with token lengths. FlashAttention is used.

9.5. Efficiency with various token length

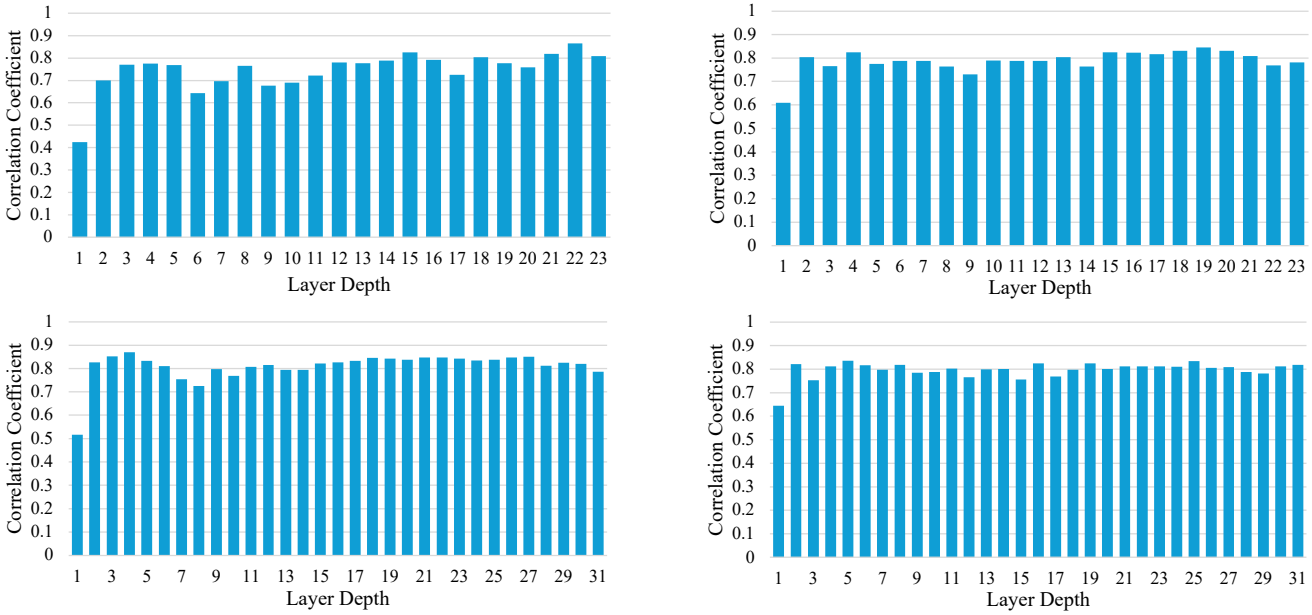
We analyze the efficiency of G-Search deployment on top of InternVL2-8B. Table 14 shows the improvement of G-Search with various token lengths. As shown, the proposed G-Search can decrease the time cost, as well as memory cost, for both prefilling and decoding with various context length, which clearly demonstrates its effectiveness in real-world deployment.

9.6. Limitations and future work

We discuss several limitations of our method and provide potential solutions. First, a general issue of prompt-aware vision token reduction methods is that they require attention scores in the prefilling stage. Thus, they cannot use existing I/O aware approaches like FlashAttention2 for further acceleration. Note that we can still use those approaches with our method in the training and the decoding stage. Although Sec. 7.2 shows that our method can outperform FlashAttention2, we will explore an I/O aware version of our method in the future. For example, we may get the calculated distances of queries and keys from static random-access memory (SRAM) to replace attention scores in our method. Second, our method finds optimal reduction strategies on image understanding data, which are not optimal for video understanding. This is probably because videos are more redundant than images in terms of vision tokens. A possible solution is to search reduction strategies on video data. Third, our method decides which token to remove only in the prefilling stage. It is highly possible that as the generation of the response, some vision tokens are no longer essential and can be removed. A potential solution is to adjust the reduction strategy based on the sequential output tokens, as well as the instruction tokens.

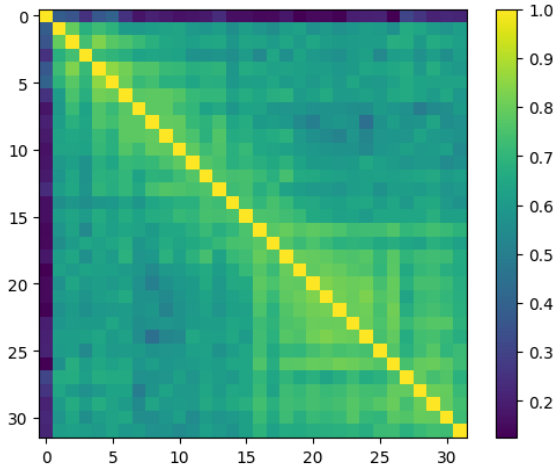


(a) LLaVA-1.5-7B

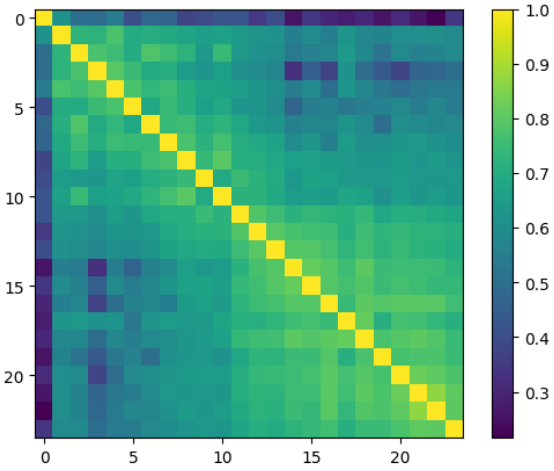


(b) InternVL2-1B (top left), 2B (top right), 4B (bottom left), 8B (bottom right)

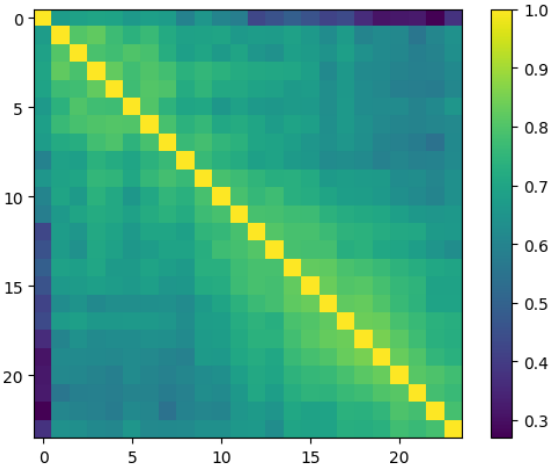
Figure 8. **(a)**: Kendall's Tau correlation coefficient between the current layer and the next layer of LLaVA-1.5-7B **(b)**: Kendall's Tau correlation coefficients for InternVL2 family. Our finding on LLaVA-1.5-7B hold on InternVL2 family.



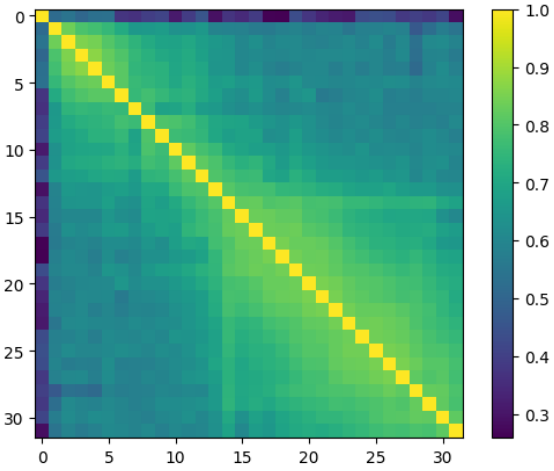
(a) LLaVA-1.5-7B



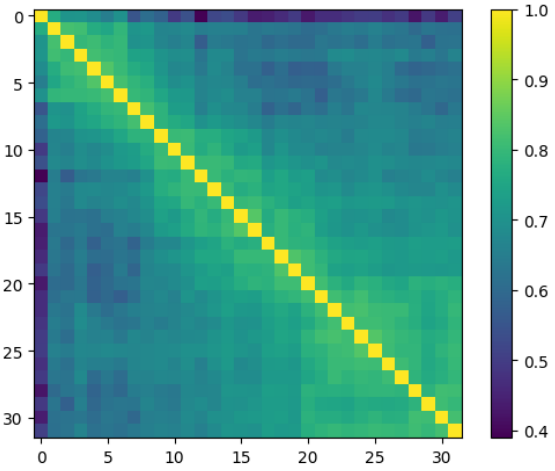
(b) InternVL2-1B



(c) InternVL2-2B



(d) InternVL2-4B



(e) InternVL2-8B

Figure 9. Kendall's Tau correlation coefficient between every two layers of various MLLMs.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. 2023. 3

[2] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301, 2019. 5

[3] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13817–13827, 2024. 1, 3

[4] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791, 2021. 4

[5] 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*, 2024. 1, 3, 6, 5

[6] 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. 2, 3, 6

[7] 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. 2, 3, 6

[8] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36, 2023. 2

[9] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 1, 2

[10] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022. 1

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 3

[12] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. 4

[13] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14375–14385, 2024. 5

[14] Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 5

[15] Yiren Jian, Chongyang Gao, and Soroush Vosoughi. Bootstrapping vision-language learning with decoupled language pre-training. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[16] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 235–251. Springer, 2016. 5

[17] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938. 1, 4, 6

[18] 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. 3

[19] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024. 4

[20] 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. 1, 2, 3, 6

[21] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 5

[22] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*, 2024. 2, 3, 4, 8

[23] Yanwei Li, Chengyao Wang, and Jiaya Jia. LLaMA-VID: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, 2024. 3

[24] 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. 1, 3

[25] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26763–26773, 2024. 1, 3

[26] Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. *arXiv preprint arXiv:2405.05803*, 2024. 1, 3, 6

[27] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2023. 2

[28] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 2, 4, 6, 8

[29] 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. 3

[30] 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? In *European Conference on Computer Vision*, pages 216–233. Springer, 2025. 5

[31] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. 5

[32] Gen Luo, Yiyi Zhou, Yuxin Zhang, Xiawu Zheng, Xiaoshuai Sun, and Rongrong Ji. Feast your eyes: Mixture-of-resolution adaptation for multimodal large language models. *arXiv preprint arXiv:2403.03003*, 2024. 1, 3

[33] Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland, 2022. Association for Computational Linguistics. 5

[34] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. DocVQA: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021. 5

[35] Jonas Mockus. The bayesian approach to global optimization. In *System Modeling and Optimization: Proceedings of the 10th IFIP Conference New York City, USA, August 31–September 4, 1981*, pages 473–481. Springer, 2005. 4

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[37] 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. 3

[38] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 5

[39] 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. 5

[40] Weiyun Wang, Shuibo Zhang, Yiming Ren, Yuchen Duan, Tiantong Li, Shuo Liu, Mengkang Hu, Zhe Chen, Kaipeng Zhang, Lewei Lu, et al. Needle in a multimodal haystack. *Advances in Neural Information Processing Systems*, 37: 20540–20565, 2024. 1, 6

[41] xAI. RealWordQA: `https://huggingface.co/datasets/xai-org/RealworldQA`, 2024. 5

[42] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3, 4, 8

[43] xiaoju ye. calflops: a flops and params calculate tool for neural networks in pytorch framework, 2023. 5

[44] 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. 1, 3, 6

[45] 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. 1, 3

[46] Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou. Deco: Decoupling token compression from semantic abstraction in multimodal large language models. *arXiv preprint arXiv:2405.20985*, 2024. 2, 3, 6, 1

[47] Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, Ying Shan, and Yansong Tang. VoCo-LLaMA: Towards vision compression with large language models. *arXiv preprint arXiv:2406.12275*, 2024. 3

[48] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024. 5

[49] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality check on the evaluation of large multimodal models, 2024. 6

[50] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011. 2

[51] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024. 4

[52] Yi-Fan Zhang, Qingsong Wen, Chaoyou Fu, Xue Wang, Zhang Zhang, Liang Wang, and Rong Jin. Beyond llava-hd: Diving into high-resolution large multimodal models. *arXiv preprint arXiv:2406.08487*, 2024. 1, 3

[53] Shiyu Zhao, Lin Zhang, Shuaiyi Huang, Ying Shen, and Shengjie Zhao. Dehazing evaluation: Real-world benchmark datasets, criteria, and baselines. *IEEE Transactions on Image Processing*, 29:6947–6962, 2020. 2

[54] Shiyu Zhao, Zhixing Zhang, Samuel Schulter, Long Zhao, BG Vijay Kumar, Anastasis Stathopoulos, Manmohan Chandraker, and Dimitris N Metaxas. Exploiting unlabeled data with vision and language models for object detection. In *European conference on computer vision*, pages 159–175. Springer, 2022. 2

[55] Shiyu Zhao, Samuel Schulter, Long Zhao, Zhixing Zhang, Yumin Suh, Manmohan Chandraker, and Dimitris N Metaxas. Taming self-training for open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13938–13947, 2024.

[56] Shiyu Zhao, Long Zhao, Yumin Suh, Dimitris N Metaxas, Manmohan Chandraker, and Samuel Schulter. Generating enhanced negatives for training language-based object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13592–13602, 2024. 2

[57] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *The Twelfth International Conference on Learning Representations*, 2024. 2