# Beyond Text-Visual Attention: Exploiting Visual Cues for Effective Token Pruning in VLMs

Supplementary Material

## Appendix

In this Appendix, we provide the following:

- Details of Experimental Setup (Appendix A)
  - Datasets
  - Model architectures
- Detailed Analysis of Attention in VLMs (Appendix B)
  - Attention distribution
  - Attention intensity
- Additional Experiments (Appendix C)
  - VisPruner with various visual encoders
  - VisPruner on background information
  - Additional ablation study
- Efficiency Analysis with FlashAttention (Appendix D)

## A. Details of Experimental Setup

### A.1. Datasets

We evaluate our method on a total of 13 widely used benchmarks, including 10 image benchmarks and 3 video benchmarks. Each task is described as follows.

#### A.1.1. Image benchmarks

We conduct experiments on 10 image benchmarks used in LLaVA [17], including 5 visual question answering benchmarks and 5 multi-modal reasoning benchmarks. All inference settings and evaluation metrics for these tasks follow the original configurations in LLaVA-1.5 [17].

**VQAv2 [5].** The VQAv2 benchmark evaluates the model's visual recognition capabilities through open-ended questions. It consists of 265,016 images from MSCOCO dataset [14], with each image containing at least 3 questions. The dataset incorporates adversarially balanced question design, ensuring that each question corresponds to at least two images with completely different answers, preventing models from relying solely on statistical patterns to derive answers. We utilize the test-dev set for evaluation, which includes 107,394 image-question pairs. Each question is associated with 10 ground truth answers, and automatic evaluation metrics are used for scoring.

**GQA [7].** The GQA benchmark focuses on evaluating structured understanding and reasoning abilities for scenes depicted in images. In addition to images and questions, it provides scene graph annotations derived from the Visual Genome dataset [10] for each image, which include structured descriptions of objects, attributes, and their relationships within the scene. The questions are generated using the scene graphs and a pre-designed engine, ensuring that each question corresponds to a clear semantic path. We use the accuracy on the test-dev set for evaluation, which contains 12,578 image-question pairs.

**VizWiz [6].** The VizWiz benchmark uses images captured by blind users to evaluate the model's visual understanding capabilities in real-world scenarios. Each image is first taken and uploaded by a blind user, accompanied by a question. The question is then paired with 10 crowdsourced answers for automated evaluation. Since the images are captured by blind users in real-life settings, some questions may be difficult to answer due to issues like blur or poor lighting. Additionally, since the images and questions originate from the same source, some questions may not be directly relevant to the image. We evaluate the model using the test-dev set, which includes 8,000 image-question pairs.

**ScienceQA [19].** The ScienceQA benchmark uses multiple-choice questions to evaluate the model's zero-shot generalization on scientific topics. The dataset contains rich domain diversity across three subjects: natural sciences, language science, and social science. Questions within each subject are hierarchically organized by topic, category, and skill, encompassing a total of 26 topics, 127 categories, and 379 skills. The images are illustrations related to the questions, and some questions do not have corresponding images. We evaluate the model using a subset of the test set that includes both questions and images, referred to as SQA-IMG, which contains 2,017 image-question pairs.

**TextVQA [21].** The TextVQA benchmark is designed to evaluate the model's ability to recognize textual information within images, emphasizing the integration of optical character recognition (OCR) and natural language understanding. The images are primarily sourced from the Open Images v3 dataset [9] and contain a variety of scenarios such as signs, billboards, and product packaging that contain rich text information. In addition to raw images, reference OCR tokens are also provided. Answers to the questions may be directly derived from the text in the images or require contextual reasoning. We evaluate the model's performance on a validation set containing 5,000 image-question pairs.

**POPE [12].** The POPE benchmark evaluates the hallucination in large vision-language models through questions about object presence. The images are sourced from the MSCOCO dataset [14], and the questions focus on whether a specific object is present in the image, assessing the degree of object hallucination. We use the average F1 score across three different sampling strategies in the test set for evaluation, including 8,910 image-question pairs.

**MME [4].** The MME benchmark aims to comprehensively evaluate the perceptual and cognitive capabilities of multimodal models, encompassing a total of 14 subtasks. The perception tasks include OCR as well as coarse- and fine-grained recognition. Coarse-grained recognition primarily focuses on the presence, count, position, and color of objects, while fine-grained recognition involves identifying specific posters, celebrities, scenes, landmarks, and artworks. All questions are binary judgment tasks. We use the perception score for performance evaluation, with 2.374 image-question pairs in total.

**MMBench [18].** The MMBench benchmark is designed to comprehensively evaluate the capabilities of multi-modal models. It defines three levels of competence from the top down, with the first level containing two basic abilities, perception and reasoning, the second level containing 6 more specific capabilities, and the third level containing 20 concrete tasks. Each task contains multiple choice questions to assess model performance on the task. The benchmark is available in both English and Chinese. The English version includes 4,377 image-question pairs, while the Chinese version, also referred to as **MMBench-CN**, contains 4,329 pairs. Both versions are used for evaluation.

**MM-Vet [24].** The MM-Vet benchmark focuses on the integration of different core vision-language capabilities. It defines 6 core capabilities, including recognition, OCR, knowledge, language generation, spatial awareness, and mathematics, which are combined into 16 specific tasks. The benchmark utilizes ChatGPT assistant for evaluation, providing unified metrics for assessing answers of varying styles. It includes a total of 218 image-question pairs.

#### A.1.2. Video benchmarks

To evaluate the performance of different methods in scenarios with higher visual redundancy, we also conduct experiments on 4 video benchmarks used in Video-LLaVA [13]. The evaluation follows Video-ChatGPT [20], using *gpt-3.5-turbo* assistant for scoring. Due to the commercial API usage limits, we follow [3] to use the first 1K samples of each benchmark in the experiments.

**TGIF-QA [8].** The TGIF-QA benchmark extends image-based VQA tasks to videos, requiring models to focus on both spatial and temporal attentions. It includes 72K animated GIFs from the Tumblr GIF dataset [11] and 165K crowdsourced question-answer pairs. We evaluate model performance using the Frame QA task in this benchmark.

**MSVD-QA [22].** The MSVD-QA benchmark is based on the Microsoft Research Video Description Corpus [2], which is commonly used for video captioning tasks. The question-answer pairs in the benchmark are derived from the descriptions in the corpus. The benchmark consists of 1,970 video clips and 50.5K question-answer pairs in total.

**MSRVTT-QA [22].** The MSRVTT-QA benchmark is based on the Microsoft Research Video to Text dataset [23], which is larger and has more complex scenes than the MSVD dataset. The benchmark consists of 10K video clips and 243K question-answer pairs in total.

### A.2. Model architectures

**LLaVA-1.5 [15].** LLaVA is one of the most widely used open-source vision-language models, and its simple design, low tuning cost, and outstanding performance make it a cornerstone in the field of multi-modal models. Specifically, LLaVA employs a pre-trained CLIP as the visual encoder and Vicuna as the text decoder. A simple linear projector connects the two modules, enabling the LLM to accept visual tokens of CLIP as input. Meanwhile, visual instruction tuning allows the model to handle vision-language tasks. Compared to the original LLaVA, LLaVA-1.5 increases the input image resolution from 224 to 336 and incorporates more instruction tuning data, resulting in a significant performance improvement.

**LLaVA-NeXT [16].** Also known as LLaVA-1.6, LLaVA-NeXT builds upon LLaVA-1.5 by further increasing the input image resolution, achieving improvements in reasoning, OCR, and world knowledge. Unlike the fixed resolution increase in LLaVA-1.5, LLaVA-NeXT employs a dynamic high-resolution design. Specifically, the model can select the best aspect ratio based on the resolution of the input image, increasing the resolution by up to 4×. Without altering the visual encoder, high-resolution images are split into several sub-images of the same size as the original image. These sub-images are individually encoded and concatenated before being fed into the LLM.

**Video-LLaVA [13].** On the basis of image understanding, Video-LLaVA extends this capability to video comprehension. It unifies representations of images and videos through alignment before projection. The overall architecture remains consistent with LLaVA: the visual encoder encodes continuous video frames individually, and the representations are concatenated as inputs to the LLM. After joint training, Video-LLaVA is capable of understanding both image and video data.

**Qwen-VL [1].** Qwen-VL is another widely used open-source vision-language model. Similar to LLaVA, it includes a visual encoder (OpenCLIP) and a text decoder (Qwen LLM). For the vision-text connector, Qwen-VL employs a vision-language adapter, which transforms image inputs into fixed-length token sequences via cross-attention. After three stages of training, Qwen-VL achieves strong vision-language understanding capabilities. And Qwen-VL-Chat is further fine-tuned based on Qwen-VL to enhance its performance in conversational tasks.

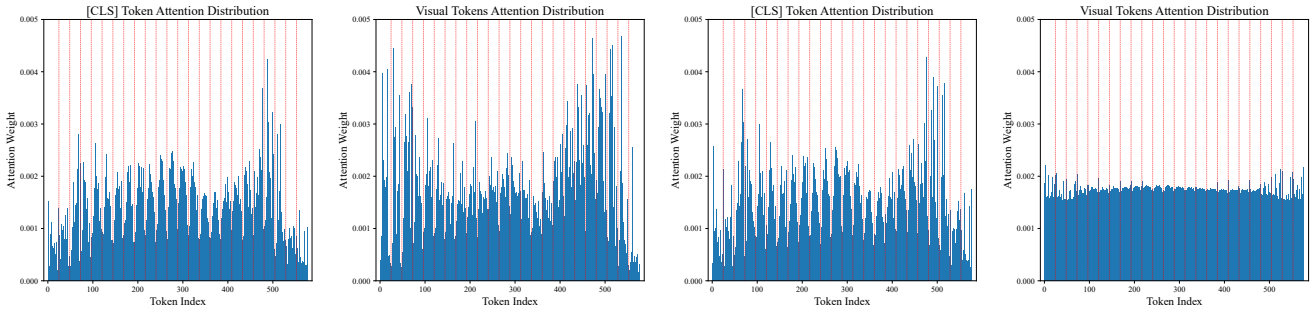


Figure A.1. Distribution of visual attention over token positions in CLIP. From left to right: attention distribution of the [CLS] token in the penultimate layer, average attention distribution across all visual tokens in the penultimate layer, attention distribution of the [CLS] token in the final layer, and average attention distribution across all visual tokens in the final layer. The red vertical dashed lines indicate the length of each row in the input image (24 for CLIP-ViT-L-14-336px used in LLaVA-1.5).

# B. Detailed Analysis of Attention in VLMs

## B.1. Attention distribution

We first present the distribution of visual attention in CLIP. As shown in Fig. A.1, the left two subplots show the visual attention in the penultimate layer of CLIP. The visual tokens used in LLaVA-1.5 also come from this layer, which retains more local features and image details. The right two subplots show the visual attention in the final layer of CLIP, which serves as the output layer. In the penultimate layer, attention from the [CLS] token is more concentrated compared to attention from other visual tokens, primarily focusing on regions closer to the image center. The [CLS] attention in the final layer is similar to that in the penultimate layer, but attention from other visual tokens becomes uniformly distributed due to the lack of supervision signals. Based on these observations, we adpot the [CLS] attention from the penultimate layer for visual token pruning.

Fig. B.1 and Fig. B.2 show the distribution of visual-text attention in the 32 layers of the 7B LLaMA language model, focusing on attention from all tokens, other visual tokens, language instruction tokens, and the last token. Unlike the attention distributions in the visual encoder, these visual-text attention distributions exhibit a clear trend of increasing intensity with larger token indices, which termed *attention shift* in the main text. Pruning visual tokens based on such attention leads to significant performance degradation, especially at high reduction ratios. This shift phenomenon is consistently observed across all types of visual-text attention. Notably, attention from the text tokens is significantly weaker than from the visual part, especially after the second layer, corroborating the inefficient visual attention phenomenon identified in FastV [3] and providing sufficient motivation for visual token pruning. Additionally, we observe that the attention distribution in the first 2 layers differs noticeably from the subsequent 30 layers. Deeper analysis of this distinction and how to leverage it to improve VLM performance is left as future work.

| Method | # Token | AI2D | ChartQA | MME | MMStar | Average |
|---|---|---|---|---|---|---|
| LLaVA-OV-7B | 729 | 80.7 | 60.5 | 1966 | 58.9 | 74.6 |
| FastV | 128 | 69.8 | 17.4 | 1397 | 43.9 | 50.2 |
| SparseVLM | 128 | 70.3 | 19.6 | 1665 | 47.0 | 55.0 |
| **VisPruner** | 128 | **74.6** | **42.8** | **1910** | **52.6** | **66.4** |

Table C.1. Performance on LLaVA-OV-7B with SigLIP encoder.

| Method | # Token | AI2D | ChartQA | MME | MMStar | Average |
|---|---|---|---|---|---|---|
| Qwen2.5-VL-7B | 1296 | 84.4 | 85.9 | 2315 | 63.8 | 87.5 |
| FastV | 256 | 77.7 | 52.0 | 2109 | 53.5 | 72.2 |
| SparseVLM | 256 | 78.4 | 54.7 | 2092 | 54.5 | 73.1 |
| **VisPruner** | 256 | **79.2** | **59.1** | **2174** | **56.8** | **76.0** |

Table C.2. Performance on Qwen2.5-VL-7B with Qwen2.5-ViT.

## B.2. Attention intensity

We visualize the attention maps from the [CLS] token in the visual encoder and the last token in the language model for the same input images in Fig. B.3. The attention from the [CLS] token is more concentrated, focusing on key foreground objects like Big Ben, train carriages, paragliders, sticky notes, the bird, giraffes, and the titles on book covers, as well as certain artifacts that encode global information. In contrast, the attention from the last token in the language model is more dispersed, spread across the entire input image. This indicates that it includes more noise, making it less effective for accurately evaluating the importance of visual tokens. This discrepancy highlights a degree of misalignment between the visual and language modalities in existing vision-language models. Addressing this misalignment to improve VLM performance on multi-modal understanding tasks remains a direction for future work.

# C. Additional Experiments

## C.1. VisPruner with various visual encoders

We apply VisPruner to LLaVA-OneVision and Qwen2.5-VL in Tabs. C.1 and C.2. The former uses SigLIP encoder, while the latter adopts a redesigned Qwen2.5-ViT en-

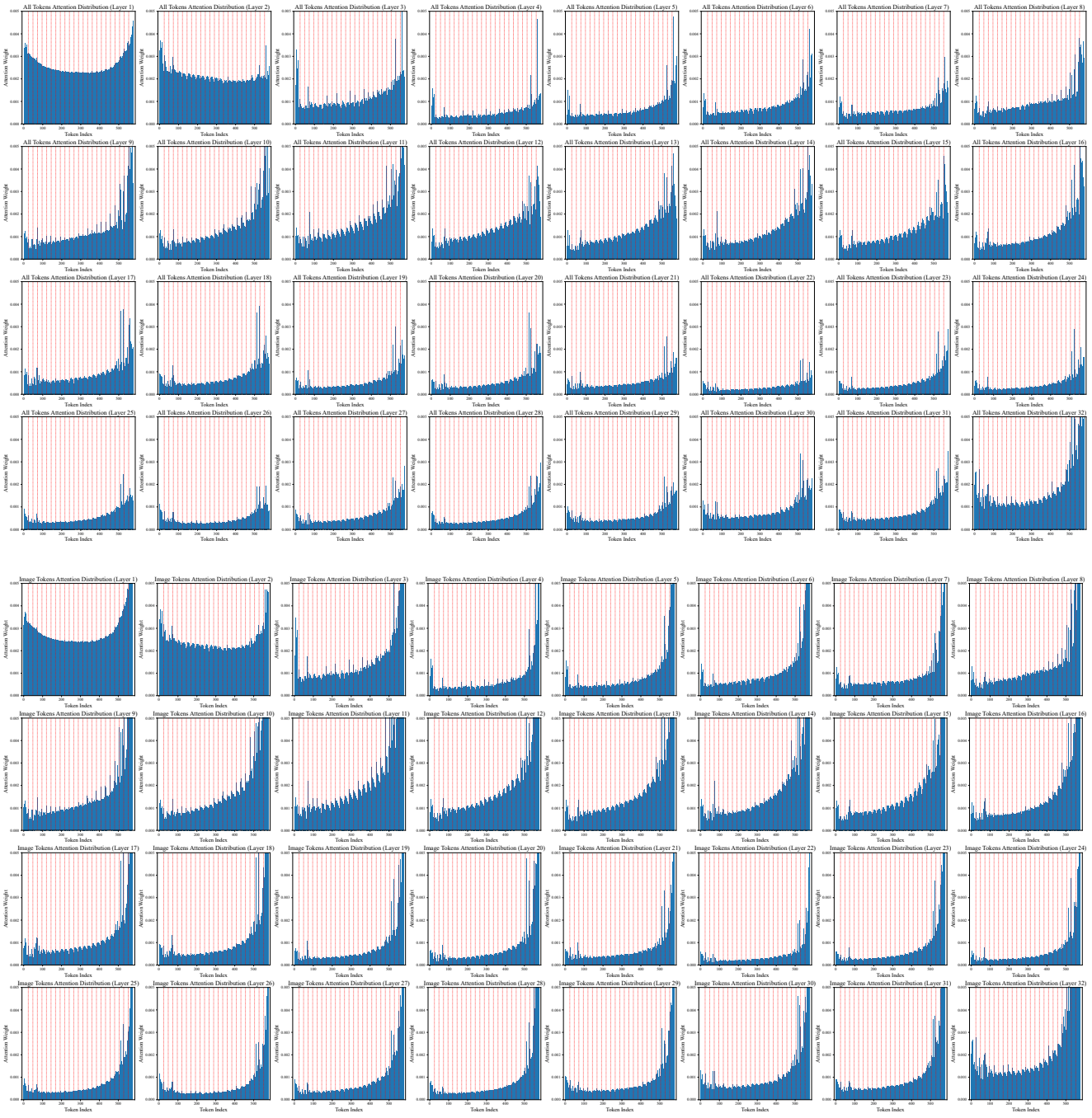


Figure B.1. Distribution of visual-text attention over visual token positions in LLaMA. The top rows display the average attention distribution across all tokens, while the bottom rows display the attention distribution from other visual tokens. Each type of attentions include results from all 32 layers of the 7B language model. The red vertical dashed lines indicate the length of each row in the input image (24 for CLIP-ViT-L-14-336px used in LLaVA-1.5).

coder. We select important tokens based on the average attention scores from all visual tokens without a [CLS] token. VisPruner achieves the best performance on both VLMs, demonstrating its effectiveness across different encoders.

## C.2. VisPruner on background information

We use GQA for this evaluation since among the five semantic types, *global* pertains to overall properties of the

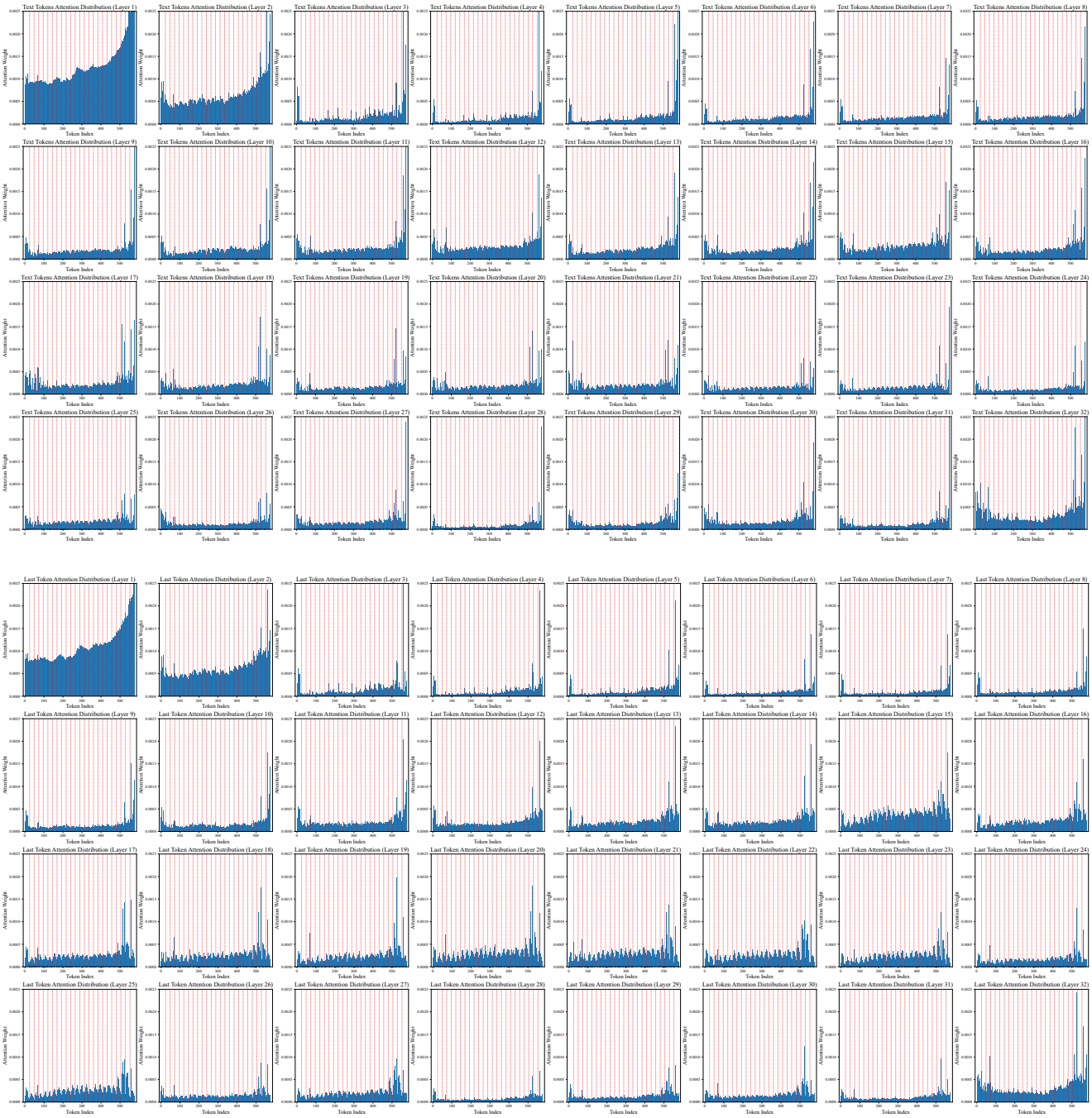


Figure B.2. Distribution of visual-text attention over visual token positions in LLaMA. The top rows display the attention distribution from language instruction tokens, while the bottom rows display the attention distribution of the last token, which is also used to predict the next token. Each type of attentions include results from all 32 layers of the 7B language model. The red vertical dashed lines indicate the length of each row in the input image (24 for CLIP-ViT-L-14-336px used in LLaVA-1.5).

scene like weather or place, requiring the model to attend to the background. As shown in Tab. C.3, incorporating diverse tokens leads to significant improvements in *global*, demonstrating its ability to retain background information.

## C.3. Additional ablation study

Here we conduct an additional ablation study on the hyperparameter $r$ of important ratio in Tabs. C.4 and C.5

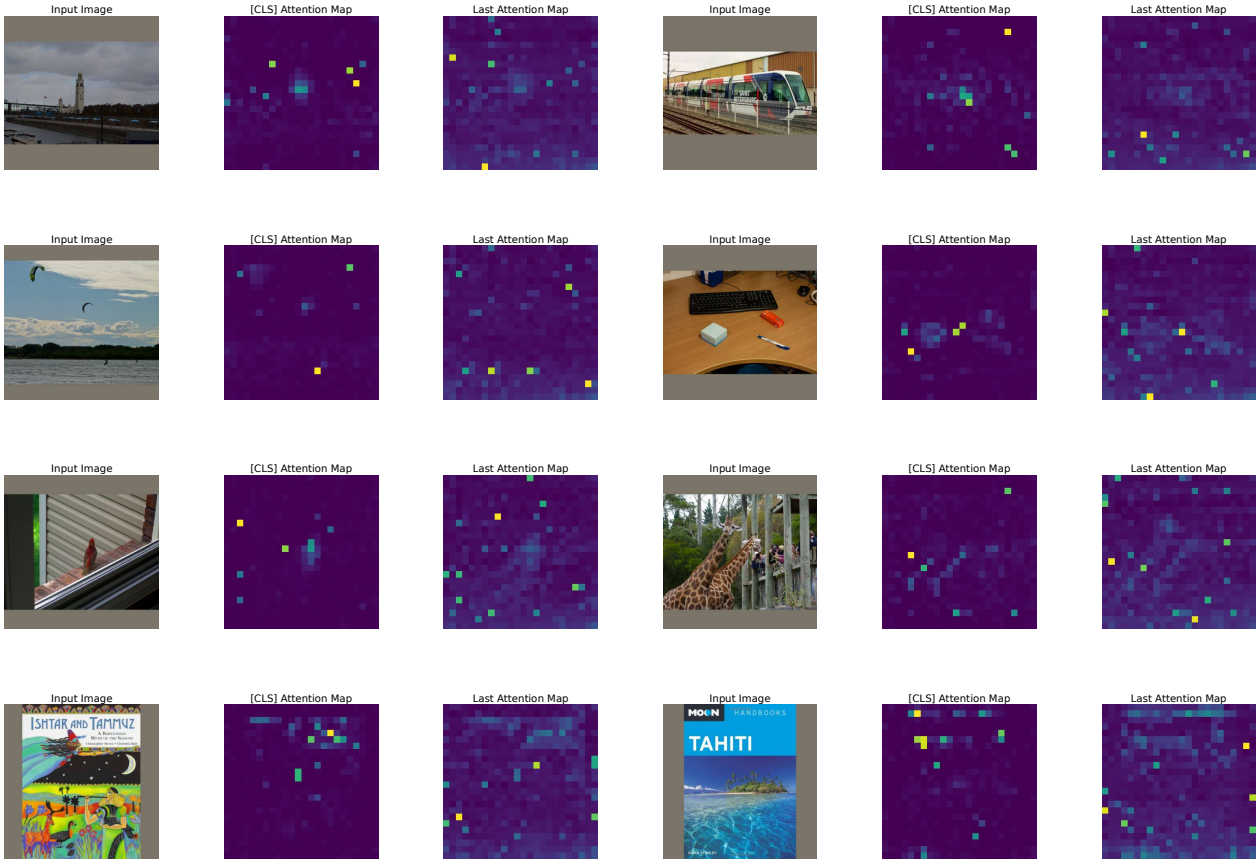


Figure B.3. Visualizations of attention maps from the [CLS] token in the visual encoder and the last token in the language model.

| Method | # Token | Five GQA Semantic Types | | | | |
|---|---|---|---|---|---|---|
| | | object | attribute | category | relation | **global** |
| LLaVA-1.5-7B | 576 | 87.7 | 68.3 | 53.2 | 53.8 | 63.7 |
| VisPruner ($r = 0.5$) | 64 | 81.9 | **60.3** | **45.4** | **48.7** | **61.2** |
| VisPruner ($r = 1.0$) | 64 | **82.3** | 59.9 | 44.7 | 48.4 | 59.2 |

Table C.3. Background information evaluation on GQA (*global*).

| $r$ | # Token | VQAV2 | GQA | TextVQA | MME | Average |
|---|---|---|---|---|---|---|
| 0.00 | 64 | 72.0 | **56.5** | 53.5 | 1324.2 | 62.1 |
| 0.25 | 64 | 72.5 | 55.8 | 55.2 | 1364.2 | 62.9 |
| **0.50** | 64 | **72.7** | 55.4 | **55.8** | **1369.9** | **63.1** |
| 0.75 | 64 | 72.6 | 55.2 | 55.6 | 1363.3 | 62.9 |
| 1.00 | 64 | 72.6 | 55.0 | 55.3 | 1355.8 | 62.7 |

Table C.4. Hyperparameter sensitivity on LLaVA-1.5-7B.

for LLaVA-1.5 and Qwen2.5-VL, respectively. The performance differences of various $r$ remain minimal. In our main experiments, we set $r$ to 0.5 to balance importance and diversity, yielding the best results on all models.

| $r$ | # Token | AI2D | ChartQA | MME | MMStar | Average |
|---|---|---|---|---|---|---|
| 0.00 | 256 | 77.5 | 58.8 | 2112 | 55.9 | 74.5 |
| 0.25 | 256 | 78.7 | 59.0 | 2127 | 56.7 | 75.2 |
| **0.50** | 256 | 79.2 | **59.1** | 2174 | 56.8 | **76.0** |
| 0.75 | 256 | 79.4 | 56.7 | 2178 | 56.8 | 75.5 |
| 1.00 | 256 | **79.6** | 54.7 | **2181** | **57.2** | 75.1 |

Table C.5. Hyperparameter sensitivity on Qwen2.5-VL-7B.

## D. Efficiency Analysis with FlashAttention

In Tabs. D.1 to D.3, we compare the computational efficiency between FastV and our VisPruner under LLaVA-1.5-7B, LLaVA-1.5-13B, and LLaVA-NeXT-7B. Unlike FastV, which prune visual token within the LLM, VisPruner prunes tokens before the LLM, enabling compatibility with FlashAttention. This design results in significantly higher efficiency. Note that the original implementation of SDPA also includes FlashAttention, so its computational efficiency is comparable to that of FlashAttention2, with only slight differences. All analyses are performed on a single NVIDIA A100-80GB GPU, evaluated on POPE.

| Method | Reduction | # Token | FLOPs (T) | Storage (MB) | GPU Memory (GB) | CUDA Time (ms) | Accuracy (%) |
|---|---|---|---|---|---|---|---|
| LLaVA-1.5-7B | 0% | 576 | 8.02 | 288.00 | 14.68 | 107.26 | 85.88 |
| FastV | 25% | 432 | 6.20 | 220.50 | 14.58 | 107.09 | 85.29 |
| VisPruner (sdpa) | | | **6.08** | **216.00** | **14.51** | 101.95 | **85.92** |
| VisPruner (flash attention) | | | | | | **101.35** | 85.87 |
| FastV | 50% | 288 | 4.40 | 153.00 | 14.52 | 99.72 | 82.45 |
| VisPruner (sdpa) | | | **4.16** | **144.00** | **14.46** | 93.57 | **86.20** |
| VisPruner (flash attention) | | | | | | **92.26** | 86.16 |
| FastV | 75% | 144 | 2.62 | 85.50 | 14.52 | 94.67 | 73.74 |
| VisPruner (sdpa) | | | **2.26** | **72.00** | **14.44** | 85.06 | **83.46** |
| VisPruner (flash attention) | | | | | | **84.03** | 83.42 |
| FastV | 90% | 58 | 1.57 | 45.31 | 14.64 | 90.48 | 57.30 |
| VisPruner (sdpa) | | | **1.13** | **29.00** | **14.54** | 79.11 | **75.85** |
| VisPruner (flash attention) | | | | | | **77.44** | 75.82 |
| FastV | 95% | 29 | 1.22 | 31.72 | 14.63 | 89.31 | 35.47 |
| VisPruner (sdpa) | | | **0.76** | **14.50** | **14.54** | 78.09 | **67.24** |
| VisPruner (flash attention) | | | | | | **77.15** | 67.22 |

Table D.1. Efficiency comparison between FastV and VisPruner under LLaVA-1.5-7B.

| Method | Reduction | # Token | FLOPs (T) | Storage (MB) | GPU Memory (GB) | CUDA Time (ms) | Accuracy (%) |
|---|---|---|---|---|---|---|---|
| LLaVA-1.5-13B | 15.28 | 450.00 | 26.98 | 156.64 | 85.99 | 107.26 | 85.88 |
| FastV | 25% | 432 | 11.69 | 343.13 | 26.61 | 151.66 | 85.86 |
| VisPruner (sdpa) | | | **11.50** | **337.50** | **26.58** | 138.36 | **86.73** |
| VisPruner (flash attention) | | | | | | **137.95** | 86.72 |
| FastV | 50% | 288 | 8.14 | 236.25 | 26.40 | 137.83 | 85.15 |
| VisPruner (sdpa) | | | **7.76** | **225.00** | **26.31** | 124.13 | **86.05** |
| VisPruner (flash attention) | | | | | | **123.33** | 86.04 |
| FastV | 75% | 144 | 4.62 | 129.38 | 26.40 | 123.69 | 79.43 |
| VisPruner (sdpa) | | | **4.05** | **112.50** | **26.29** | 104.81 | **83.10** |
| VisPruner (flash attention) | | | | | | **103.57** | 83.09 |
| FastV | 90% | 58 | 2.53 | 65.70 | 26.40 | 114.98 | 67.26 |
| VisPruner (sdpa) | | | **1.86** | **45.31** | **26.27** | 94.80 | **74.71** |
| VisPruner (flash attention) | | | | | | **94.04** | 74.66 |
| FastV | 95% | 29 | 1.83 | 44.19 | 26.55 | 114.25 | 49.83 |
| VisPruner (sdpa) | | | **1.12** | **22.66** | **26.43** | 94.48 | **65.90** |
| VisPruner (flash attention) | | | | | | **93.73** | 65.79 |

Table D.2. Efficiency comparison between FastV and VisPruner under LLaVA-1.5-13B.

# References


[1] 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. 3

[2] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011. 3

[3] 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, 2024. 3, 4

[4] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2024. 3

[5] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 2

[6] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018. 2

[7] 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. 2

[8] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and

| Method | Reduction | # Token | FLOPs (T) | Storage (MB) | GPU Memory (GB) | CUDA Time (ms) | Accuracy (%) |
|---|---|---|---|---|---|---|---|
| LLaVA-NeXT-7B | 0% | 2880 | 43.58 | 1440.00 | 17.04 | 313.04 | 86.77 |
| FastV | 25% | 2160 | 33.05 | 1102.50 | 16.95 | 262.49 | 86.63 |
| VisPruner (sdpa) | | | **32.35** | **1080.00** | **16.33** | 246.38 | **86.69** |
| VisPruner (flash attention) | | | | | | **232.07** | 86.66 |
| FastV | 50% | 1440 | 23.04 | 765.00 | 16.95 | 201.68 | 85.73 |
| VisPruner (sdpa) | | | **21.66** | **720.00** | **15.11** | 176.80 | **87.06** |
| VisPruner (flash attention) | | | | | | **170.30** | 87.02 |
| FastV | 75% | 720 | 13.53 | 427.50 | 16.95 | 147.93 | 82.68 |
| VisPruner (sdpa) | | | **11.51** | **360.00** | **14.80** | 119.87 | **86.50** |
| VisPruner (flash attention) | | | | | | **116.79** | 86.46 |
| FastV | 90% | 290 | 8.10 | 226.56 | 16.95 | 117.33 | 70.77 |
| VisPruner (sdpa) | | | **5.71** | **145.00** | **14.70** | 87.36 | **81.17** |
| VisPruner (flash attention) | | | | | | **85.26** | 81.12 |
| FastV | 95% | 145 | 6.30 | 158.59 | 16.95 | 112.19 | 49.36 |
| VisPruner (sdpa) | | | **3.80** | **72.50** | **14.70** | 78.18 | **74.77** |
| VisPruner (flash attention) | | | | | | **77.66** | 74.72 |

Table D.3. Efficiency comparison between FastV and VisPruner under LLaVA-NeXT-7B.


Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2758–2766, 2017. 3

[9] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2017. 2

[10] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 2

[11] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. Tgif: A new dataset and benchmark on animated gif description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4641–4650, 2016. 3

[12] 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. 2

[13] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 3

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2

[15] 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. 3

[16] 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

[17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 2

[18] 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. 3

[19] Pan Lu, Swaroop Mishra, Tanglin 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. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022. 2

[20] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023. 3

[21] 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. 2

[22] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653, 2017. 3

[23] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016. 3

[24] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023. 3