VisionZip: Longer is Better but Not Necessary in Vision Language Models

Senqiao Yang[1] Yukang Chen[1] Zhuotao Tian[3*] Chengyao Wang[1] Jingyao Li[1] Bei Yu[1] Jiaya Jia[1,2]
[1]CUHK [2]HKUST [3]HITSZ

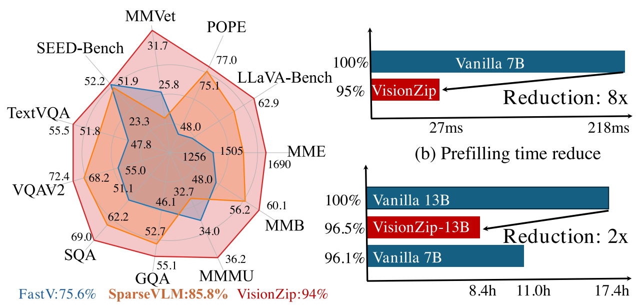
Abstract

Recent advancements in vision-language models have enhanced performance by increasing the length of visual tokens, making them much longer than text tokens and significantly raising computational costs. However, we observe that the visual tokens generated by popular vision encoders, such as CLIP and SigLIP, contain significant redundancy. To address this, we introduce VisionZip, a simple yet effective method that selects a set of informative tokens for input to the language model, reducing visual token redundancy and improving efficiency while maintaining model performance. The proposed VisionZip can be widely applied to image and video understanding tasks and is well-suited for multi-turn dialogues in real-world scenarios, where previous methods tend to underperform. Experimental results show that VisionZip outperforms the previous state-of-the-art method by at least 5% performance gains across nearly all settings. Moreover, our method significantly enhances model inference speed, improving the prefilling time by 8× and enabling the LLaVA-Next 13B model to infer faster than the LLaVA-Next 7B model while achieving better results. Furthermore, we analyze the causes of this redundancy and encourage the community to focus on extracting better visual features rather than merely increasing token length. Our code is available at https://github.com/dvlab-research/VisionZip.

1. Introduction

Recently, the advancement of Large Language Models (LLMs) [1, 2, 51, 72] has led to significant progress in Vision Language Models (VLMs) [3, 5, 27, 31, 33]. To integrate visual signals with textual semantics, existing VLMs typically utilize sequential visual representation, where images are converted into vision tokens and processed by an LLM decoder. Through modal alignment and instruction tuning, these VLMs adapt LLMs to the vision domain, leveraging their perception and reasoning capabilities.

However, the promising performance of VLMs largely relies on the large amount of visual tokens. For example, in LLaVA-1.5 [33], the number of visual tokens is 576, and in LLaVA-NeXT [34], a 672x672 image yield more than 576x5=2880 tokens, while the text tokens number only in the dozens to just over a hundred. These excessively long visual tokens consume a significant amount of memory and computation in the entire VLM, limiting the model's development in practical application scenarios such as edge computing, autonomous driving, and robotics [23, 36, 41, 57, 59, 60]. Furthermore, based on many previous studies [1, 4, 10, 22], we know that the information contained in images is much sparser than in text. In contrast, the existing state-of-the-art VLMs have far more visual tokens than text tokens. Hence, a natural question arises: ***"Are all visual tokens necessary?"***



(a) VisionZip outperforms sota EfficientVLM (c) Boost 13B faster and better than 7B

Figure 1. **VisionZip Performance and Efficiency.** (a) Our VisionZip significantly outperforms the current SOTA EfficientVLM model, like FastV, SparseVLM, achieving nearly 95% of the performance with only 10% of the tokens across 11 benchmarks on LLaVA-1.5. (b) VisionZip could reduce 8× prefilling time for LLaVA-NeXT 7B. (c) VisionZip reduces GPU inference time by 2× across 11 benchmarks, enabling the LLaVA-NeXT 13B model to infer faster than the 7B model while achieving better results.

To explore this, we conduct a pilot study on the visual tokens generated by the widely used vision encoders, CLIP [42] and SigLIP [64]. As shown in Fig. 2, statistical and visual analysis reveal that only a few tokens receive high attention and contain a large amount of information, while most visual tokens receive minimal attention and aggregate limited information. Based on the observation, we can answer the question that ***there is a significant amount of redundancy in the visual tokens.*** Details of this phe-

*Corresponding author (tianzhuotao@hit.edu.cn)

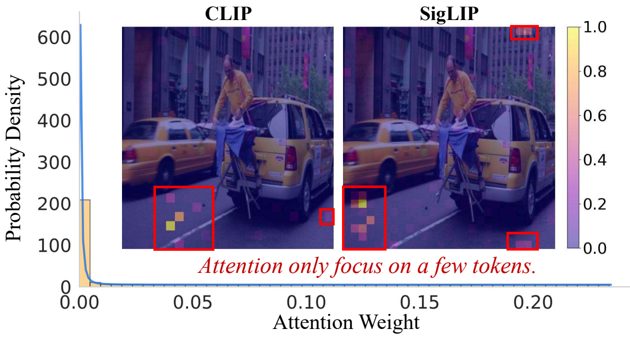


Figure 2. **Redundancy Visualization.** The visualization and distribution statistics of attention scores show attention concentrated on only a few tokens, while many tokens display very low attention scores, indicating significant redundancy in the visual tokens.

nomenon's observation and the reasons behind it are provided in Sec. 2.2 and Sec. 4.1, respectively.

Based on this observation, we explore a solution to reduce visual token redundancy, aiming to improve efficiency without sacrificing performance. Specifically, we develop a text-agnostic method named VisionZip to extract more informative visual tokens for the LLM. Our method can be used in training-free, fine-tuning, or training from scratch. Specifically, in training-free mode, we first select the dominant tokens, which receive significant attention and aggregate most of the image information. Then, to avoid missing small but potentially important details, we employ a token merging strategy, merging retained tokens based on their similarity to further extract informative contextual tokens. In fine-tuning mode (in Sec. 2.4), after selecting tokens to replace all raw visual tokens, the input token count decreases significantly, leading to a slight misalignment between the current visual input space and the LLM space. To enhance results and improve alignment, we fine-tune the projector layer for 30 minutes with minimal data, enabling the model to adapt to the reduced token count.

To demonstrate the effectiveness of our method, we apply the proposed VisionZip to popular VLM models and evaluate it on several benchmarks in Sec. 3. As shown in Fig. 1, the results indicate that even in a training-free scenario, our method significantly outperforms previous state-of-the-art methods in both speed and performance. Furthermore, VisionZip can reduce pre-filling time by 8 times while retaining 95% performance in LLaVA-NeXT 7B. The proposed VisionZip also enables LLaVA-NeXT 13B to achieve better performance and faster inference than the LLaVA-NeXT 7B model. Finally, we analyze the causes of the redundancy and explain why the simple, text-agnostic VisionZip achieves better performance than previous methods, highlighting its advantages in real-world deployment like multi-turn conversations in Sec. 4.

Algorithm 1 Pseudocode for Dominant Token Selection

```
# B: batch size; S: sequence length
# H: number of attention heads;
# K: number of target dominant tokens
# CLS_IDX: Index of the CLS token
# SELECT_LAYER: Selected layer for Visual Token

# set the output_attentions=True to get the attention
output = vision_tower(images, output_hidden_states=
    True, output_attentions=True)

#attn in shape (B, H, S, S)
attn = output.attentions[SELECT_LAYER]

#attn in shape (B, H, S, S)
vanilla_tokens = output.hidden_states[SELECT_LAYER]

#The attention received by each token
#If no CLS, use mean calculate received attention
attn_rec = attn[:, :, cls_idx, cls_idx+1:].sum(dim=1)

# Select K Dominant Tokens
_, topk_idx = attn_rec.topk(K, dim=1)

# Concat with the CLS token
dominant_idx = cat(CLS_IDX, topk_idx+1)

# filter the Dominant Tokens
dominant_tokens = vanilla_tokens.filter(dominant_idx)
```

`cat`: concatenation; `filter`: select the tokens based on the index.

2. VisionZip

In this section, we first explain the importance of reducing the number of visual tokens to improve model efficiency in Sec. 2.1, and then present our observation of redundancy in Sec. 2.2. After that, we detail the training-free method in Sec. 2.3. Additionally, to help the model better adapt to variations in visual token length, we introduce Efficient Tuning in Sec. 2.4. Finally, we briefly discuss the widespread usage of VisionZip. The overall architecture is shown in Fig. 3.

2.1. Preliminary

Architecture of VLM. The VLM architectures generally consist of three components: a visual encoder, a modality projector, and a LLM. The visual encoder, typically a pre-trained image encoder like CLIP's vision model, converts input images into visual tokens. The projector module aligns these visual tokens with the LLM's word embedding space, enabling the LLM to process visual data effectively. The LLM then integrates the aligned visual and textual information to generate responses.

Computation Complexity. Evaluating the computational complexity of VLMs requires examining key components such as the self-attention mechanism and the feed-forward network (FFN). The total floating-point operations (FLOPs) can be expressed as:

$$\text{Total FLOPs} = T \times (4nd^2 + 2n^2d + 2ndm)$$

where T is the number of transformer layers, n is the sequence length, d is the hidden dimension size, and m represents the intermediate size of the FFN.

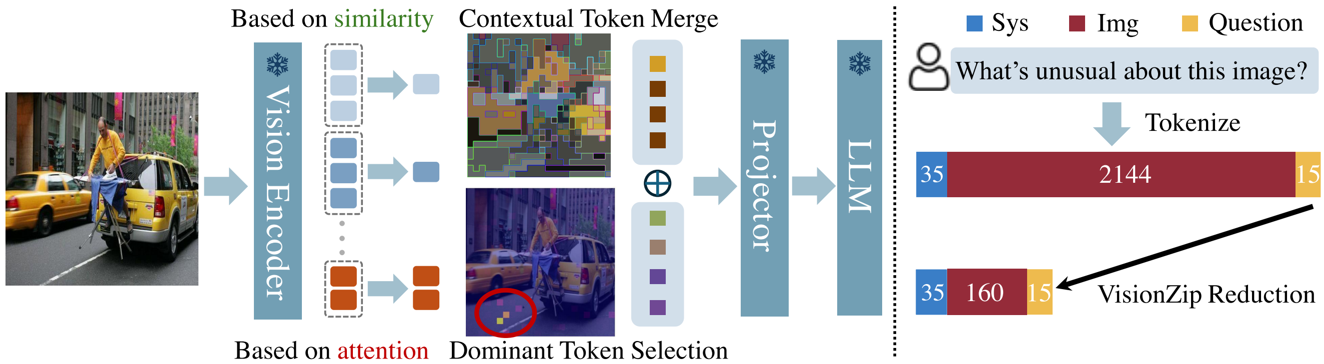


Figure 3. **Framework of VisionZip.** VisionZip selects dominant tokens that aggregate substantial information based on visual token attention scores. Remaining tokens are merged based on semantic similarity to produce contextual tokens. VisionZip is a training-free method significantly reduces the number of image tokens, accelerating inference while maintaining performance. With efficient fine-tuning of the projector, even better results can be achieved with minimal performance loss compared to using the full token.

This equation shows that computational complexity is strongly influenced by the sequence length n. In typical VLM tasks, the sequence length is defined as $n = n_{\text{sys}} + n_{\text{img}} + n_{\text{question}}$, with n_{img} often being much larger than the other two, sometimes by a factor of 20. Thus, **reducing n_{img}** is essential for improving the efficiency of VLMs.

2.2. Redundancy Observation

In popular Vision Language Models like LLaVA and Mini-Gemini, the number of vision tokens far exceeds that of text tokens, consuming substantial computational resources. To assess whether all these tokens are necessary, we conducted a pilot study on the visual tokens generated by commonly used vision encoders, CLIP and SigLIP.

Specifically, we randomly sampled one image and visualized the attention of each token from the Vision Encoder's -2 layer, which is the selected layer for obtaining input visual tokens in most VLMs, such as the LLaVA. As shown in Fig. 2, both CLIP and SigLIP exhibit an attention pattern concentrated on a limited number of tokens, while the majority of visual tokens receive minimal attention. Furthermore, to demonstrate that the attention focusing on only a few tokens is a normal phenomenon, we analyze the distribution of attention weights on the TextVQA validation set. As shown in Fig. 2, most visual tokens receive very low attention, with weights close to zero, while only a few tokens hold higher attention weights. To show this phenomenon's prevalence, we include more visualizations in Appendix D.1.

Based on this observation, we find that most visual tokens with low attention weights contribute little information and add significant redundancy. Only a few visual tokens aggregate a substantial amount of information and merit focused attention; we refer to these as the dominant visual tokens. Therefore, to reduce redundancy, we focus on selecting the most informative tokens—such as the dominant visual tokens—while discarding less informative ones to reduce the overall token count.

Algorithm 2 Pseudocode for Contextual Tokens Merging.

```
# Remove dominant tokens
remaining = vanilla_tokens.mask(dominant_tokens)

# Split into target and merge tokens
# M represents the desired number of contextual tokens
targets, merge = uniform_split(remaining, M)

# Compute similarity based on the key values
simlarity = bmm(to_merge.K, targets.K.transpose(1, 2))

# Assign each merge token to the most similar target
assign_idx = simlarity.argmax(dim=2)

# Merge by averaging
context_tokens = avg_merge(assign_idx, targets, merge)
```

`uniform_split`: Uniformly sample the target tokens, and the rest are the merge tokens; `avg_merge`: Average merge the tokens based on the assigned indices.

2.3. Informative Visual Token Zip

Dominant Token Selection. To reduce redundancy by retaining only the most informative visual tokens and discarding less significant ones, the main challenge is identifying which tokens contribute most to the model's performance. We evaluate the importance of each visual token by examining its attention scores within the vision encoder. Specifically, we calculate the attention score as Eq. 1,

$$\boldsymbol{S}_h = \text{Softmax}\left(\frac{\boldsymbol{Q}_h \boldsymbol{K}_h^\top}{\sqrt{D_h}}\right), \tag{1}$$

where $\boldsymbol{S}_h$ is the attention score of each head, D_h is the head dimension, and $\boldsymbol{Q}_h$ and $\boldsymbol{K}_h$ represent query and key, respectively. Averaging across the head dimension, yields an aggregated attention matrix $\boldsymbol{S}_{avg} \in \mathbb{R}^{B \times SeqLen \times SeqLen}$, reflecting how each token attends to others.

For models with a CLS token, such as CLIP, which aggregates information from the entire image, we leverage the

CLS token's attention scores to identify key visual tokens. As shown in Algorithm 1, we select the tokens most attended to by the CLS token, as these typically contain the most relevant information. For models without a CLS token, such as SigLIP, we calculate the average attention each token receives from all others in the sequence. Tokens with higher average attention are considered more significant and retained. We provide the details of it in Appendix A.2.

This process allows us to efficiently identify and retain the dominant visual tokens, as shown in Fig. 3, these tokens contain almost all attention and aggregate substantial information of the total tokens.

Contextual Tokens Merging. Although we have selected dominant tokens by evaluating their significance, and these dominant tokens contain most visual information, we merge the remaining tokens to avoid losing any small but potentially important information. Specifically, during self-attention calculation, the keys (K) already summarize the information contained in each token. Therefore, as shown in Algorithm 2, we first uniformly split the non-dominant tokens into target and merge tokens. We then use a similarity metric, such as the dot product, to identify the keys containing similar information. Finally, we merge the tokens that contain the most similar information, creating contextual tokens. As shown in Fig. 3, these contextual tokens serve as highly informative tokens, containing the figure's semantic similarity information.

2.4. Efficient Tuning

The Informative Visual Token Zip extracts highly informative tokens from the visual encoder and drops other tokens, thereby significantly reducing the token length input to the LLM, potentially by up to tenfold. However, this reduction in visual tokens can lead to a degree of misalignment, as the VLM model, originally trained on all full visual tokens, may struggle to adapt to the sudden decrease.

To bridge the gap between the visual and LLM spaces, we use minimal instruction tuning data to efficiently finetune the multimodal projector while keeping other components frozen, enhancing alignment between the vision and language spaces. Notably, the instruction tuning requires only 1/10 of the LLaVA-1.5 dataset and can be completed in just 30 minutes on 8 Nvidia A800 for LLaVA 1.5 7B. Notably, this process can also be implemented on 3090 GPUs, which is both resource-efficient and effective.

2.5. Usage of VisionZip

The VisionZip can adapt to multiple tasks, not only for image and video understanding in Vision-Language Models but also for multi-turn conversations that previous efficient VLMs could not handle. Additionally, VisionZip is easy to implement as it is text-agnostic, enabling compatibility with all existing LLM algorithms for acceleration. Furthermore, VisionZip can be seen as a plug-and-play method for vision encoders, which preserves over 90% of the original model's performance while saving 3 times runtime and memory. It can even allow a 13B VLM to achieve greater efficiency than a 7B VLM while maintaining superior performance. We will show more details in Sec. 4.3.

3. Experiments

3.1. Effectiveness on Image Understanding

Evaluation Tasks. To show the effectiveness of our method on image understanding tasks, we conduct experiments on eleven widely used benchmarks [11, 13, 19, 26, 29, 33, 37, 39, 47, 61, 63] and compare our method with the existing sota methods, FastV [6] and SparseVLM [66], which progressively reduce the number of visual tokens in the LLM forward process based on attention weights. To further validate the generalizability of our method, we conduct experiments on various VLM with different architectures and resolutions. Due to space limitations, we present only a subset of results for LLaVA-1.5 [33], LLaVA-NeXT [34], and Mini-Gemini [31] in the main text and all results and implementation details can be found in Appendix B.

Results on LLaVA 1.5. As shown in Table 1, we deploy the proposed VisionZip on LLaVA-1.5 and demonstrate its performance on image understanding tasks. VisionZip represents our method being directly applied during the inference stage without additional training. VisionZip‡ denotes an efficient tuning for the cross-modality projector, requiring approximately 30 minutes on 8 A800 GPUs. This tuning can also be implemented on 3090 GPUs, making it both resource-efficient and effective. To comprehensively assess performance, we present the results in percentage format for comparative analysis, with the vanilla model's accuracy serving as the 100% upper limit. Following the setup in [6, 66], we use three vision token count configurations (192, 128, and 64) to evaluate the advantages of our proposed VisionZip. When the visual tokens are reduced from 576 to 192, VisionZip only decreases the average accuracy by 1.5% without additional training, surpassing FastV [6] by 10.3% and SparseVLM [66] by 2.1%, respectively. Furthermore, when only 64 tokens remain, our method outperforms FastV [6] and SparseVLM [66] by a significant margin of **18.4%** and **8.2%**, respectively. Additionally, VisionZip‡, which efficiently tunes the cross-modality projector, provides further performance improvements. As shown in Table 1, even with only 64 visual tokens retained, this efficient tuning boosts performance to 95.2%, representing only a 4.8% decrease compared to the vanilla method using 10 times the visual tokens.

An interesting phenomenon is that in certain benchmarks, such as MMVeT and MMMU, using VisionZipto reduce the token count not only prevents performance degradation but also improves performance. We believe the rea-

Method	GQA	MMB	MME	POPE	SQA	VQA^{V2}	VQA^{Text}	MMMU	SEED	MMVet	LLaVA-B	Avg.
Upper Bound, 576 Tokens **(100%)**												
Vanilla (CVPR24)	61.9 100%	64.7 100%	1862 100%	85.9 100%	69.5 100%	78.5 100%	58.2 100%	36.3 100%	58.6 100%	31.1 100%	66.8 100%	100%
Retain 192 Tokens (↓ **66.7**%)												
FastV (ECCV24)	52.7 85.1%	61.2 94.6%	1612 86.6%	64.8 75.4%	67.3 96.8%	67.1 85.5%	52.5 90.2%	34.3 94.5%	57.1 97.4%	27.7 89.7%	49.4 74.0%	88.2%
SparseVLM (2024.10)	57.6 93.1%	62.5 96.6%	1721 92.4%	83.6 97.3%	69.1 99.4%	75.6 96.3%	56.1 96.4%	33.8 93.1%	55.8 95.2%	31.5 101.3%	66.1 99.0%	96.4%
VisionZip	59.3 95.8%	63.0 97.4%	1782.6 95.7%	85.3 99.3%	68.9 99.1%	76.8 97.8%	57.3 98.5%	36.6 100.8%	56.4 96.2%	31.7 101.9%	67.7 101.3%	**98.5%**
VisionZip ‡	60.1 97.1%	63.4 98.0%	1834 98.5%	84.9 98.8%	68.2 98.1%	77.4 98.6%	57.8 99.3%	36.2 99.7%	57.1 97.4%	32.6 104.8%	66.7 99.9%	**99.1%**
Retain 128 Tokens (↓ **77.8**%)												
FastV (ECCV24)	49.6 80.1%	56.1 86.7%	1490 80.0%	59.6 69.4%	60.2 86.6%	61.8 78.7%	50.6 86.9%	34.9 96.1%	55.9 95.4%	28.1 90.9%	52.0 77.8%	83.5%
SparseVLM (2024.10)	56.0 90.5%	60.0 92.7%	1696 91.1%	80.5 93.7%	67.1 96.5%	73.8 94.0%	54.9 94.3%	33.8 93.1%	53.4 91.1%	30 96.5%	62.7 93.9%	93.4%
VisionZip	57.6 93.1%	62.0 95.8%	1761.7 94.6%	83.2 96.9%	68.9 99.1%	75.6 96.3%	56.8 97.6%	37.9 104.4%	54.9 93.7%	32.6 104.8%	64.8 97.6%	**97.6%**
VisionZip ‡	58.9 95.2%	62.6 96.8%	1823 97.9%	83.7 97.4%	68.3 98.3%	76.6 97.6%	57.0 97.9%	37.3 102.8%	55.8 95.2%	32.9 105.8%	64.8 97.0%	**98.4%**
Retain 64 Tokens (↓ **88.9**%)												
FastV (ECCV24)	46.1 74.5%	48.0 74.2%	1256 67.5%	48.0 55.9%	51.1 73.5%	55.0 70.1%	47.8 82.1%	34.0 93.7%	51.9 88.6%	25.8 83.0%	46.1 69.0%	75.6%
SparseVLM (2024.10)	52.7 85.1%	56.2 86.9%	1505 80.8%	75.1 87.4%	62.2 89.4%	68.2 86.9%	51.8 89.0%	32.7 90.1%	51.1 87.2%	23.3 74.5%	57.5 86.1%	85.8%
VisionZip	55.1 89.0%	60.1 92.9%	1690 90.8%	77.0 89.6%	69.0 99.3%	72.4 92.2%	55.5 95.4%	36.2 99.7%	52.2 89.1%	31.7 101.9%	62.9 94.2%	**94.0%**
VisionZip ‡	57.0 92.1%	61.5 95.1%	1756 94.3%	80.9 94.2%	68.8 99.0%	74.2 94.5%	56.0 96.2%	35.6 98.1%	53.4 91.1%	30.2 97.1%	63.6 95.2%	**95.2%**

Table 1. **Performance of VisionZip on LLaVA 1.5.** The vanilla number of visual tokens is 576. The first line of each method shows the raw benchmark accuracy, and the second line is the proportion relative to the upper limit. The last column is the average value. VisionZip‡ indicates that fine-tuning the multimodal projector with $1/10$ LLaVA-1.5 datasets, which takes 30 minutes for 8A800 GPU.

son is that the visual tokens are overly redundant, and this redundant information not only fails to improve model performance but may also act as noise, impacting the model's judgment and leading to performance degradation. We analyze this phenomenon in Sec. 4.

Results on LLaVA-NeXT. To further demonstrate the effectiveness of our proposed VisionZip, we apply it to the more advanced, high-resolution-capable VLM, LLaVA-NeXT. Compared to LLaVA 1.5, LLaVA-NeXT divides the image into four parts, resizes the original image, and converts it into five separate images. Each of these images is processed through the visual encoder to obtain visual tokens, which are then combined. While this approach further improves model performance, it significantly increases the number of visual tokens. Therefore, to enhance efficiency, we aim to use our method to reduce the number of visual tokens as much as possible without compromising model performance. And we set the three vision token count configurations (640, 320, and 160) to evaluate the advantages of our proposed VisionZip. As shown in Table 2, our proposed VisionZip consistently maintains strong performance across three settings. Specifically, using only 640 tokens, our method achieves 97.6% accuracy without any additional training cost. With minimal data used to tune the projector, VisionZip's performance reaches 98.9%, which is very close to that of the vanilla model. Additionally, when the visual token count is reduced to only about 5%, our method still achieves 92.0% performance without any ad-

Method	GQA	MMB	MME	SQA	VQA^{V2}	VQA^{Text}	MMMU	Avg.
Upper Bound, 2880 Tokens **(100%)**								
Vanilla	64.2 100%	67.9 100%	1842 100%	70.2 100%	80.1 100%	61.3 100%	35.1 100%	100%
Retain 640 Tokens (↓ **77.8%**)								
SparseVLM	60.3 93.9%	65.7 96.8%	1772 96.2%	67.7 96.4%	77.1 96.3%	57.8 94.3%	34.6 98.6%	96.1%
VisionZip	61.3 95.5%	66.3 97.6%	1787 97.0%	68.1 97.0%	79.1 98.8%	60.2 98.2%	34.7 98.9%	**97.6%**
VisionZip ‡	62.4 97.2%	65.9 97.1%	1778 96.5%	67.9 96.7%	79.9 99.8%	60.8 99.2%	37.2 106.0%	98.9%
Retain 320 Tokens (↓ **88.9%**)								
SparseVLM	57.7 89.9%	64.3 94.7%	1694 92.0%	67.3 95.9%	73.4 91.6%	55.9 91.2%	34.4 98.0%	93.3%
VisionZip	59.3 92.3%	63.1 92.9%	1702 92.4%	67.3 95.9%	76.2 95.1%	58.9 96.1%	35.3 100.5%	**95.0%**
VisionZip ‡	61.0 95.0%	64.4 94.8%	1770 96.1%	67.5 96.2%	78.4 97.9%	59.3 96.7%	38.0 108.3%	97.9%
Retain 160 Tokens (↓ **94.4%**)								
SparseVLM	51.2 79.8%	63.1 92.9%	1542 83.7%	67.5 96.2%	66.3 82.8%	46.4 75.7%	32.8 93.4%	86.4%
VisionZip	55.5 86.4%	60.1 88.5%	1630 88.5%	68.3 97.3%	71.4 89.1%	56.2 91.7%	36.1 102.8%	**92.0%**
VisionZip ‡	58.2 90.7%	63.9 94.1%	1699 92.2%	67.5 96.2%	75.6 94.4%	57.3 93.5%	37.7 107.4%	95.5%

Table 2. **Performance of VisionZip on LLaVA-NeXT.** The vanilla number of visual tokens is 2880. For VisionZip‡, we use 1/10 LLaVA-1.5 datasets to fine-tune the multimodal projector.

Method	TGIF	MSVD	MSRVTT	ActivityNet	Avg
Video-LLaVA	47.1	69.8	56.7	43.1	100.0%
FastV	23.1 49.0%	38.0 54.4%	19.3 34.0%	30.6 71.0%	52.1%
SparseVLM	44.7 94.9%	68.2 97.7%	31.0 54.7%	42.6 98.8%	86.5%
VisionZip	42.4 90.0%	63.5 91.0%	52.1 91.9%	43.0 99.8%	**93.2%**

Table 3. **Performance of VisionZip on Video-LLaVA.** The original Video-LLaVa's video token number is 2048, while our VisionZip only retain the 136 tokens.

ditional training and reaches 95.2% after tuning, surpassing the previous state-of-the-art method, SparseVLM [66], by 5.8% and 9%, respectively. And the full experiment results can be found in Appendix B.

Results on Mini-Gemini. We have verified the effectiveness of our method on the LLaVA Family VLMs, and we further validate our proposed VisionZip on Mini-Gemini, which introduces a LAION-pretrained ConvNeXt-L [38] for high-resolution refinement, to demonstrate VisionZip's effectiveness across different architectures. As shown in Fig. 4, we visualize the performance change across different visual token counts on POPE, TextVQA, and GQA. It can be observed that as the number of tokens decreases, the gap between our method and the previous sota method increases sharply. These results further verify the effectiveness of our method across various model architectures and demonstrate the presence of visual token redundancy across multiple architectures. We discuss in Section 4 why our straightforward and easy-to-implement method VisionZip outperforms previous approaches.

3.2. Effectiveness on Video Understanding

Evaluation Tasks. We evaluate our method on four common video question-answering benchmarks: TGIF-QA [20], MSVD-QA [56], MSRVTT-QA [56], and ActivityNet-QA [62], where video-question pairs exhibit significant length disparities. We follow the evaluation framework proposed by Video-LLaVA [32], utilizing ChatGPT score as key performance metrics. Further details are provided in Appendix B.

Results on Video-LLaVA. The vanilla Video-LLaVA [32] uses the Language-bind as vision encoder to encode 8 frames, with each frame containing 256 visual tokens, resulting in a total of 2048 visual tokens. Hence, we set the Video-LLaVA with 2048 video tokens as the upper bound, achieving an overall average accuracy of 100.0% and a score of 0.00. To make a fair comparison, we follow the original settings for the baseline methods FastV [6]and SparseVLM [66], pruning the visual tokens to 135. For each frame, we zip the visual tokens from 256 to 17, resulting in a total of 136 visual tokens for the entire video. As shown in Table 3, our VisionZip in training-free mode achieves 93.2% accuracy across four benchmarks, outperforming the previous state-of-the-art method, SparseVLM, by 6.7%. Moreover, on the largest benchmarks, MSRVTT, our method shows a significant improvement over SparseVLM by 37.2%. Additionally, our method consistently exceeds 90% performance across all benchmarks, further demonstrating VisionZip's effectiveness and robustness.

3.3. Efficiency Analysis

Our proposed VisionZip reduces the number of visual tokens input to the Large Language Model, resulting in significant efficiency and CUDA memory gains during inference. We conduct a comparative analysis of CUDA memory usage, and pre-filling time on LLaVA NeXT-7B, comparing our method with FastV [6], and SparseVLM [66].

As shown in Table 4, we perform an inference efficiency analysis on a single NVIDIA A800-80GB, using POPE[29] dataset a fair comparison. "Prefilling time" refers to the latency required to generate the first token. The results show that our method not only surpasses previous approaches in

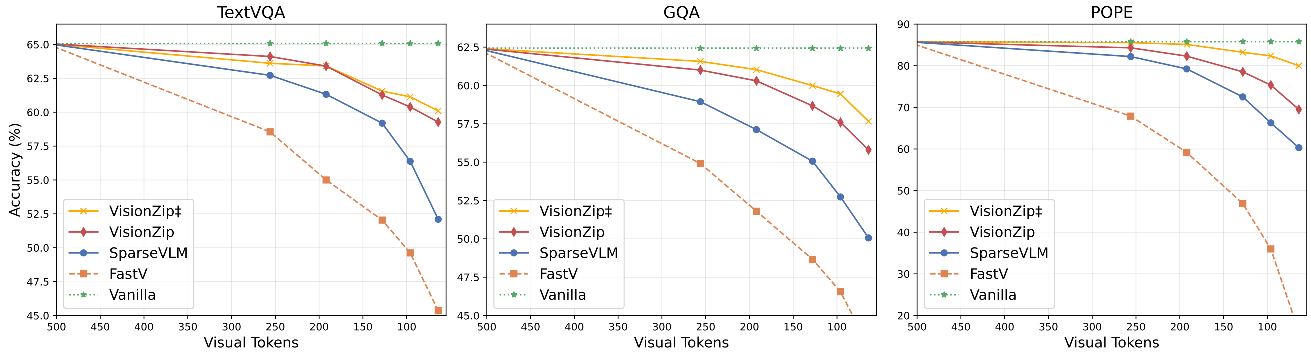


Figure 4. Performance of VisionZip on the Mini-Gemini.

Method	Token	Total Time↓	Δ	Prefilling Time ↓	Δ
Baseline	2880	2293s	-	218ms	-
FastV	160	1792s	1.3×	119ms	1.8×
SparseVLM	160	1895s	1.2×	128ms	1.7×
VisionZip	160	**756s**	**3.0**×	**27.8ms**	**7.8**×

Table 4. **Efficiency analysis of VisionZip on LLaVA-NeXT 7B.** The detailed metrics include practical total time for one A800 GPU on POPE, Prefilling time(latency). Δ denotes the reduction ratio.

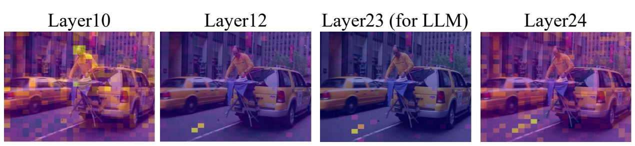


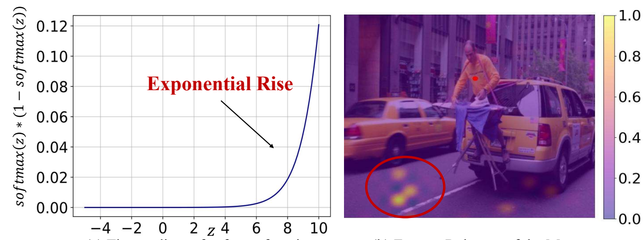
Figure 5. Visualization of attention distribution across layers

performance but also maintains a substantial advantage over previous sota methods when reduced to the same number of tokens. On the POPE dataset, our method achieves a **3**× improvement in overall time efficiency and a **7.8**× improvement in prefilling time compared to the vanilla model.

4. Analysis and Discussion

4.1. Reasons of Redundancy in Visual Tokens

Visualization of the Redundancy. Firstly, as shown in Fig. 5, we illustrate attention changes across layers. In early layers, attention is broadly distributed across the image, but by the middle layers, it suddenly converges onto a few tokens. With deeper layers, attention and information concentrate on a small set of dominant tokens, reaching peak concentration by the 23rd layer—used for visual token extraction for the LLM. Notably, attention is more dispersed in the final layer, as these tokens align with the CLIP text branch via contrastive loss, potentially limiting their representation of the original image. This is why VLM selects the second-to-last layer (-2 layer). Additional visualization



(a) The gradient of softmax function (b) Feature Relevant of the Man

Figure 6. Reason of redundancy and feature misalignment

results are in Appendix D.

Explanation. Current vision encoders are based on a transformer architecture that aggregates information between tokens through self-attention. We think that as the layer depth increases, instead of aggregating knowledge from all tokens, the model tends to "shortcut" by concentrating information into a few proxy tokens. If a CLS token is present, the knowledge may further concentrate from these proxy tokens into the CLS token. Moreover, using the function $\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$ to compute the model's loss can intensify this effect. The derivative of this formula is as:

$$\frac{\partial \text{softmax}(z_i)}{\partial z_i} = \text{softmax}(z_i) \cdot (1 - \text{softmax}(z_i)) \quad (2)$$

We illustrated this function in Fig. 6 (a), when z is large, the gradient becomes substantial in exponential rise, and when z is small, the gradient is almost negligible. This function makes regions of low attention even lower and high-attention areas even more prominent, ultimately concentrating information into a few tokens. [54] identified a similar phenomenon in LLM inference, naming it "Attention Sink." [45] also observed a comparable effect in semantic segmentation, referring to it as the "global token."

	Token	Accuracy	Δ
Baseline	576→64	51.1	
$Ex1$	526→64	46.4	−9.2%
$Ex2$	128→64	52.5	+2.7%

Table 5. Quantitative analysis for the feature misalignment

4.2. Why VisionZip Outperforms Previous Work?

Text-Relevant Efficient VLM. Existing sota methods for reducing visual redundancy to accelerate VLMs, such as FastV [6] and SparseVLM [66], primarily rely on the LLM to identify text-relevant visual token. Specifically, they feed all visual tokens into the LLM and use attention between text and visual tokens across LLM layers for selection.

Misalignment Due to the Pre-group Knowledge. While the text-relevant method appears promising, the visual tokens it selects often lack sufficient information. This limitation arises because the visual encoder aggregates visual information into a limited subset of high-attention tokens, leaving the remaining tokens with minimal informational content. As a result, tokens that should represent specific details are instead grouped into proxy tokens, losing their original incontext information. Furthermore, these proxy tokens tend to appear in peripheral or background areas rather than being positioned near the main subjects of the image. For instance, in Fig. 6 (b), the visual tokens most relevant to the person are not located on the person but are instead assigned to a proxy token situated on the road. This indicates that text-relevant methods often select tokens from elements like the man or the taxi, which actually contain significantly less informative content.

To further verify this, we performed two experiments on the TextVQA benchmark with SparseVLM, retaining 64 tokens, as shown in Table 5. In $Ex1$, we first masked 50 out of 576 total tokens, selecting the 50 tokens with the highest attention according to the vision encoder. From the remaining 526 tokens, SparseVLM was used to select the final set. This approach reduced performance from 51.1 to 46.4, a drop of approximately 9%. In $Ex2$, instead of providing all 576 tokens, we only supplied the top 128 tokens selected by VisionZip to SparseVLM, which then filtered down to the final 64 tokens. This approach improved performance to 52.5, an increase of about 2.6%. These results further verify that the text-relevant visual tokens are misaligned with the tokens where the Vision Encoder aggregates knowledge.

4.3. The Advantage of the VisionZip

Advantage on multi-turn conversations. To better support real-world applications, current VLMs store the previous answer in the KV cache to enable multi-turn conversations, reducing the need to reprocess prior dialogue. However, as shown in Figure 7, prior text-relevant methods are unsuitable for multi-turn conversations. This is because the visual tokens selected and stored in the KV cache are closely related to the previous question but lack relevance to the current dialogue, leading to poor performance in multi-turn scenarios. In contrast, our VisionZip selects the most informative visual tokens in a text-agnostic manner, making it more effective for multi-turn conversations. Other advantages are shown in the Appendix A.3.

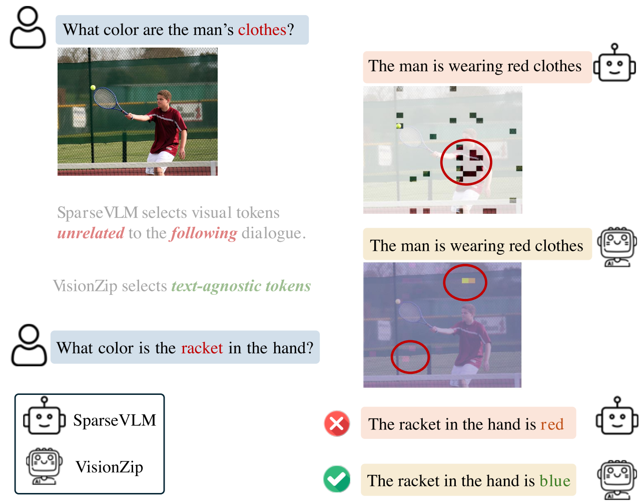


Figure 7. Example comparison of VisionZip and previous text-relevant method in multi-turn conversation

5. Related Work

Vision-Language Models. Building on the success of large language models (LLMs) [1, 2, 51], recent vision-language models (VLMs) [8, 31, 33, 50] advance multi-modal generation by processing extensive visual token sequences. Higher resolutions require exponentially more tokens; for example, LLaVA-NeXT processes 672×672 images into 2304 tokens [33]. Handling videos or multiple images increases token requirements, as seen in VideoLLaVA [32] and Video-ChatGPT [40]. Hence, it's essential to discuss more efficient ways to extract information from visual tokens, rather than merely increasing their length. The additional related work is shown in Appendix C.

6. Conclusion

In this paper, we analyze popular VLM models, noting that while increasing the length of visual tokens can improve performance, there is significant redundancy in current visual tokens. We propose a simple method, VisionZip, which reduces the number of visual tokens substantially while preserving model performance, thereby greatly enhancing computational efficiency. This method is broadly applicable to image and video understanding tasks and is suitable for multi-turn dialogue in practical applications.VisionZip also suggests a future direction to develop vision encoders with lower redundancy capabilities to further improve VLM performance and handle longer video sequences.

7. Acknowledgments

This work was supported by the Research Grants Council under the Areas of Excellence scheme grant AoE/E-601/22-R and Shenzhen Science and Technology Innovation Program (KJZD20240903102901003).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv:2303.08774*, 2023. 1, 8, 20

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv:2309.16609*, 2023. 1, 8, 20

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A frontier large vision-language model with versatile abilities. *arXiv:2308.12966*, 2023. 1

[4] Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56, 2023. 1

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

[6] 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, 2024. 4, 6, 8, 12, 20

[7] Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. In *ICLR*, 2024. 20

[8] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv:2312.14238*, 2023. 8

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023. 20

[10] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[11] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv:2306.13394*, 2023. 4, 15

[12] Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive kv cache compression for llms. *arXiv preprint arXiv:2310.01801*, 2023. 20

[13] 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. 4, 15

[14] 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. 14

[15] Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Simple on-the-fly length generalization for large language models. *arXiv preprint arXiv:2308.16137*, 2023. 20

[16] Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipvl: Efficient large vision-language models with dynamic token sparsification and kv cache compression. *arXiv preprint arXiv:2410.08584*, 2024. 12, 20

[17] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. Lita: Language instructed temporal-localization assistant. In *European Conference on Computer Vision*, pages 202–218. Springer, 2025. 20

[18] Zhengchao Huang, Bin Xia, Zicheng Lin, Zhun Mou, and Wenming Yang. Ffaa: Multimodal large language model based explainable open-world face forgery analysis assistant. *arXiv preprint arXiv:2408.10072*, 2024. 20

[19] Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4, 15

[20] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and 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. 6, 18

[21] Yiren Jian, Tingkai Liu, Yunzhe Tao, Chunhui Zhang, Soroush Vosoughi, and Hongxia Yang. Expedited training of visual conditioned language generation via redundancy reduction. *arXiv preprint arXiv:2310.03291*, 2023. 20

[22] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, page 2. Minneapolis, Minnesota, 2019. 1

[23] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 1

[24] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 20

[25] Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv:2406.18629*, 2024. 20

[26] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking mul-

timodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 4, 13

[27] 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*, 2023. 1

[28] Jingyao Li, Han Shi, Xin Jiang, Zhenguo Li, Hong Xu, and Jiaya Jia. Quickllama: Query-aware inference acceleration for large language models, 2024. 20

[29] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv:2305.10355*, 2023. 4, 6, 15

[30] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. 2024. 20

[31] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv:2403.18814*, 2024. 1, 4, 8, 13, 20

[32] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv:2311.10122*, 2023. 6, 8, 20

[33] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv:2310.03744*, 2023. 1, 4, 8, 13, 14, 16, 20

[34] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 1, 4, 13, 20

[35] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 2024. 20

[36] Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Lily Lee, Kaichen Zhou, Pengju An, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Multimodal state space model for efficient robot reasoning and manipulation. *arXiv preprint arXiv:2406.04339*, 2024. 1

[37] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. MMBench: Is your multi-modal model an all-around player? *arXiv:2307.06281*, 2023. 4, 14

[38] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 6

[39] 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. 4, 14

[40] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 2024. 8, 20

[41] Guanqiao Qu, Qiyuan Chen, Wei Wei, Zheng Lin, Xianhao Chen, and Kaibin Huang. Mobile edge intelligence for large language models: A contemporary survey. *arXiv preprint arXiv:2407.18921*, 2024. 1

[42] 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. 1, 12

[43] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024. 20

[44] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024. 20

[45] Tong Shao, Zhuotao Tian, Hang Zhao, and Jingyong Su. Explore the potential of clip for training-free open vocabulary semantic segmentation. In *European Conference on Computer Vision*, pages 139–156. Springer, 2025. 7

[46] Dachuan Shi, Chaofan Tao, Ying Jin, Zhendong Yang, Chun Yuan, and Jiaqi Wang. Upop: Unified and progressive pruning for compressing vision-language transformers. In *International Conference on Machine Learning*, pages 31292–31311. PMLR, 2023. 20

[47] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019. 4, 15

[48] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024. 20

[49] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, et al. Video understanding with large language models: A survey. *arXiv preprint arXiv:2312.17432*, 2023. 20

[50] 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. 8, 20

[51] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv:2302.13971*, 2023. 1, 8, 20

[52] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan

Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv:2311.03079*, 2023. 20

[53] Yuxin Wen, Qingqing Cao, Qichen Fu, Sachin Mehta, and Mahyar Najibi. Efficient vision-language models by summarizing visual tokens into compact registers. *arXiv preprint arXiv:2410.14072*, 2024. 20

[54] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv*, 2023. 7, 20

[55] 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. 12, 20

[56] 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 ACM international conference on Multimedia*, pages 1645–1653, 2017. 6, 18

[57] Senqiao Yang, Jiaming Liu, Ray Zhang, Mingjie Pan, Zoey Guo, Xiaoqi Li, Zehui Chen, Peng Gao, Yandong Guo, and Shanghang Zhang. Lidar-llm: Exploring the potential of large language models for 3d lidar understanding. *arXiv preprint arXiv:2312.14074*, 2023. 1

[58] Senqiao Yang, Tianyuan Qu, Xin Lai, Zhuotao Tian, Bohao Peng, Shu Liu, and Jiaya Jia. An improved baseline for reasoning segmentation with large language model. *arXiv preprint arXiv:2312.17240*, 2023. 20

[59] Senqiao Yang, Zhuotao Tian, Li Jiang, and Jiaya Jia. Unified language-driven zero-shot domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23407–23415, 2024. 1

[60] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 1

[61] 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. In *International conference on machine learning*. PMLR, 2024. 4, 14

[62] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, pages 9127–9134, 2019. 6, 18

[63] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024. 4, 13

[64] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 1

[65] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*, 2024. 14, 15, 16, 17, 19, 20

[66] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024. 4, 6, 8, 12, 20, 21

[67] Yuechen Zhang, Shengju Qian, Bohao Peng, Shu Liu, and Jiaya Jia. Prompt highlighter: Interactive control for multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13215–13224, 2024. 20

[68] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36: 34661–34710, 2023. 20

[69] Chuanyang Zheng, Yihang Gao, Han Shi, Minbin Huang, Jingyao Li, Jing Xiong, Xiaozhe Ren, Michael Ng, Xin Jiang, Zhenguo Li, and Yu Li. Dape: Data-adaptive positional encoding for length extrapolation, 2024. 20

[70] Qiang Zhou, Chaohui Yu, Shaofeng Zhang, Sitong Wu, Zhibing Wang, and Fan Wang. Regionblip: A unified multimodal pre-training framework for holistic and regional comprehension. *arXiv preprint arXiv:2308.02299*, 2023. 20

[71] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852*, 2023. 12

[72] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv:2304.10592*, 2023. 1