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, 49, 71] has led to significant progress in Vision Language Models (VLMs) [3, 5, 26, 30, 32, 55]. 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 for 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 exam-
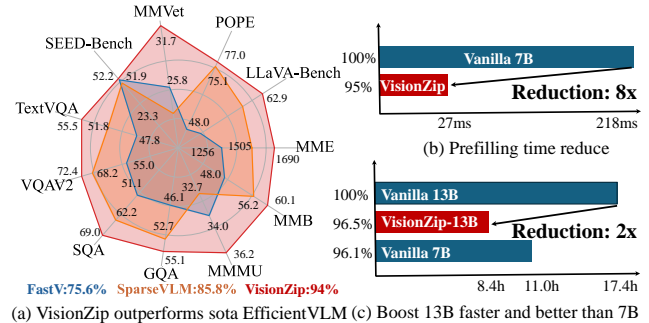

Figure 1. **VisionZip Performance and Efficiency.** (a) Our VisionZip significantly outperforms the current SOTA EfficientVLM model, like FastV, SparseVLM, achieving nearly 95% of the performance with only 10% of the tokens across 11 benchmarks on LLaVA-1.5. (b) VisionZip could reduce 8× prefilling time 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.

ple, in LLaVA-1.5 [32], the number of visual tokens is 576, and in LLaVA-NeXT [33], 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 [22, 35, 40, 56, 58, 59]. Furthermore, based on many previous studies [1, 4, 10, 21], 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?"*

To explore this, we conduct a pilot study on the visual tokens generated by the widely used vision encoders, CLIP [41] and SigLIP [63]. 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-
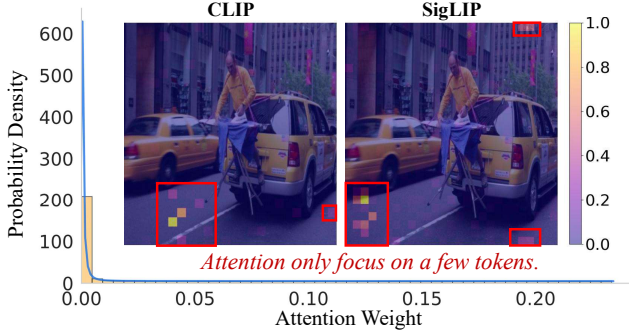
1

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 repre-
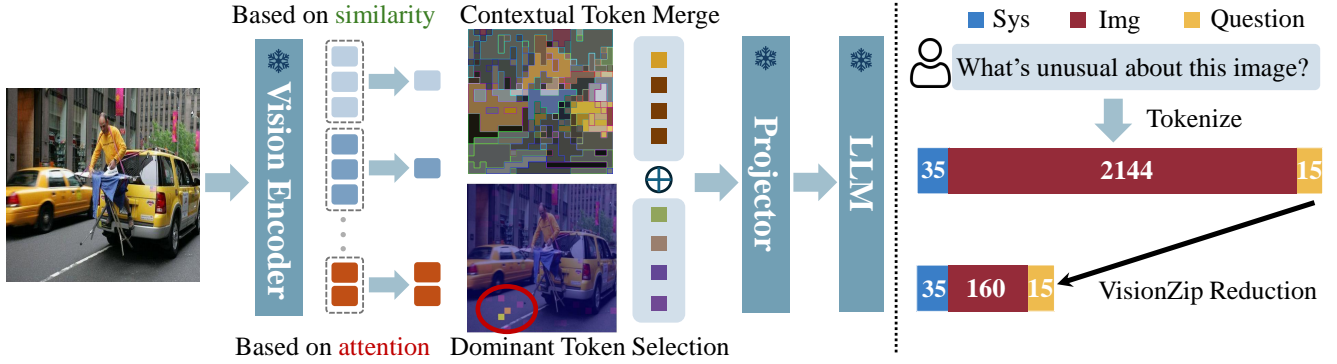
2

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.

sents the intermediate size of the FFN.

This equation shows that computational complexity is strongly influenced by the sequence length n. In typical VLM tasks, the sequence length is defined as $n = n_{sys} + n_{img} + n_{question}$, 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 se-

---

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.

lecting the most informative tokens—such as the dominant visual tokens—while discarding less informative ones to reduce the overall token count.

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,

$$S_h = \text{Softmax}\left(\frac{Q_h K_h^\top}{\sqrt{D_h}}\right), \qquad (1)$$

where S_h is the attention score of each head, D_h is the head dimension, and Q_h and K_h represent query and key, respectively. Averaging across the head dimension, yields an aggregated attention matrix $S_{avg} \in \mathbb{R}^{B \times SeqLen \times SeqLen}$, 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 fine-tune 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, 25, 28, 32, 36, 38, 45, 60, 62] and compare our method with the existing sota methods, FastV [6] and SparseVLM [65], 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 [32], LLaVA-NeXT [33], and Mini-Gemini [30] 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, 65], 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 [65] by 2.1%, respectively. Furthermore, when only 64 tokens remain, our method outperforms FastV [6] and SparseVLM [65] 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 bench-

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

marks, such as MMVeT and MMMU, using VisionZip to reduce the token count not only prevents performance degradation but also improves performance. We believe the reason 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 con-

verts 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

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

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.

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 additional training and reaches 95.2% after tuning, surpassing the previous state-of-the-art method, SparseVLM [65], 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 [37] 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

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

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.

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 [54], MSRVTT-QA [54], and ActivityNet-QA [61], where video-question pairs exhibit significant length disparities. We follow the evaluation framework proposed by Video-LLaVA [31], utilizing Chat-GPT score as key performance metrics. Further details are provided in Appendix B.

Results on Video-LLaVA. The vanilla Video-LLaVA [31] 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 [65], 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 us-
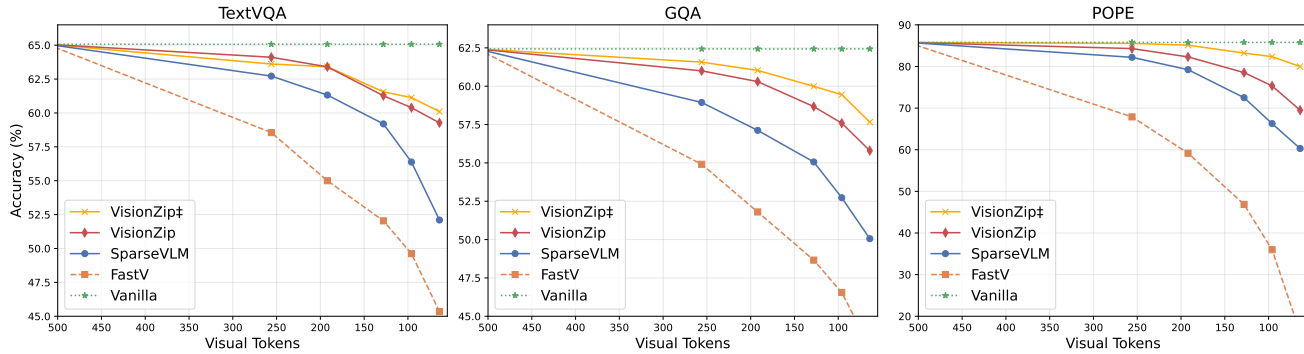
Figure 4. Performance of VisionZip on the Mini-Gemini.

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

age, and pre-filling time on LLaVA NeXT-7B, comparing our method with FastV [6], and SparseVLM [65].

As shown in Table 4, we perform an inference efficiency analysis on a single NVIDIA A800-80GB, using POPE[28] 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 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 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. [52] identified a similar phenomenon in LLM inference, naming it "Attention Sink." [43] also observed a comparable effect in se-
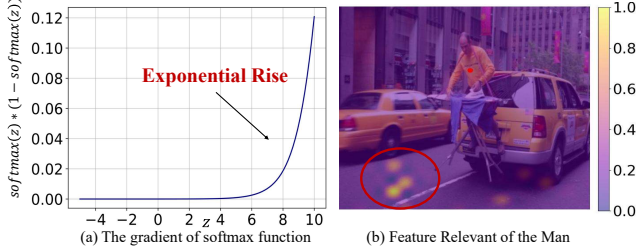
7

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

Figure 6. Reason of redundancy and feature misalignment

mantic 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 [65], 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

Easy to deployment. Due to VisionZip directly reducing the visual tokens before projecting them into the LLM, rather than gradually reducing them during the LLM forward process, it avoids extensive computation and memory consumption in the LLM's shallow layers. As shown in Table 6, our method is compatible with existing quantization techniques, maintaining performance while minimizing memory usage. Furthermore, our method enables the 13B model to be faster and perform better than the 7B model. As shown in Table 7, our method significantly reduces the inference time of the 13B model, making it twice as fast as the vanilla 13B model and outperforming the vanilla 7B model in both performance and efficiency. Full results across 11 evaluation benchmarks are provided in Appendix B. Additionally, VisionZip is well-suited for integration with LLM acceleration optimization algorithms. **Advantage**

Precision	Memory	Acc
7B-Full	18,952	70.2
13B-Full	36,721	73.5
13B-8bit-†	16,632	70.8
13B-4bit-†	**10,176**	**70.3**

Table 6. Compatibility of VisionZip on various quantization levels for ScienceQA. † represents use of VisionZip.

Size	Time	Acc
7B	1,714s	61.3
13B	2,516s	64.3
13B†	**1,246s**	**62.2**

Table 7. VisionZip boosts the 13B model's performance and efficiency over the 7B model on TextVQA. † represents use of VisionZip.

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.
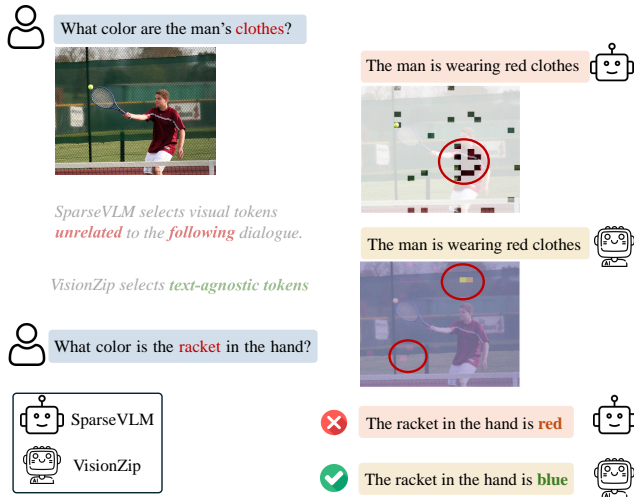
8

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, 49], recent vision-language models (VLMs) [8, 30, 32, 48] 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 [32]. Handling videos or multiple images increases token requirements, as seen in VideoLLaVA [31] and Video-ChatGPT [39]. 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.

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, 9, 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, 9, 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, 7, 8, 12, 21

[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. 9

[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, 14

[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, 14

[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, 21

[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, 14

[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, 17

[21] 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

[22] 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

[23] 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

[24] 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 preprint arXiv:2406.18629*, 2024. 20

[25] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 4, 13

[26] 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

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

[28] 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, 7, 14

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

[30] 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, 9, 14, 20

[31] 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, 9, 20

[32] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv:2310.03744*, 2023. 1, 4, 9, 14, 15, 20

[33] 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, 14, 20

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

[35] 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

[36] 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

[37] 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

[38] 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

[39] 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. 9, 20

[40] 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

[41] 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, 13

[42] 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

[43] 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

[44] 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. 21

[45] 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, 14

[46] 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

[47] 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

[48] 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. 9, 20

[49] 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, 9, 20

[50] 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

[51] 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. 21

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

[53] 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, 21

[54] 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, 17

[55] Fuzhao Xue, Yukang Chen, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Ethan He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. Longvila: Scaling long-context visual language models for long videos. *CoRR*, abs/2408.10188, 2024. 1

[56] 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

[57] 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

[58] 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

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

[60] 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, 13

[61] 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, 17

[62] 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

[63] 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

[64] 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. 15, 16, 17, 18, 19, 20

[65] 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, 7, 8, 12, 21

[66] 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

[67] 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

[68] 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

[69] 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

[70] 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. 13

[71] 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

Contents

A. Further Discussion

A.1. Comparison with Text-relevant Efficient VLM

We observe that most recent Efficient VLMs [6, 16, 53, 65] utilize attention mechanisms between text tokens and visual tokens to determine which visual tokens should be retained, processing them during the LLM forward. However, our method, VisionZip, removes visual token redundancy before inputting them into the LLM. We will demonstrate our advantages from the following perspectives.

Better Performance. As shown in Table 1, 2, 3 of the main paper, our VisionZip achieves better performance in the training-free mode. This is because the Vision Encoder pre-groups the visual information into a few tokens, which often appear in the background or less prominent areas. However, when tokens are selected based on the semantic information of the text, the chosen tokens are often not the dominant tokens and carry less information, resulting in lower performance compared to VisionZip. Additionally, to better demonstrate the misalignment caused by the Vision Encoder's pre-grouping of information, we have created an interactive demo. As shown in Fig. 15, the code for this demo will be published soon.

More Efficient. Our method reduces the redundancy of visual tokens before inputting them into the LLM, avoiding the heavy attention computation in the early layers of the LLM (Sec. B.3). Additionally, we observe that previous text-relevant Efficient VLMs require significant intermediate computations to determine which tokens need to be dropped during the LLM forward process. This leads to a noticeable increase in memory usage, sometimes exceeding that of the vanilla model. This issue is particularly evident in models like LLaVA-NeXT, where the number of visual tokens is substantial.

More Application Scenarios. VisionZip operates out-

side the LLM, making it compatible with any existing LLM and applicable to all acceleration algorithms designed for LLMs. Furthermore, VisionZip is better suited for practical applications such as multi-turn conversations and other real-world scenarios.

A.2. VisionZip for Non-CLS Vision Encoders

Although most popular vision encoders, such as CLIP [41], OpenCLIP, and LanguageBind [70], use the CLS token to aggregate information, a recently introduced vision encoder, SigLIP, does not include the CLS token. To demonstrate the generalization of our proposed VisionZip, we explain how to apply it to Non-CLS Vision Encoders in this section.

Specifically, for the Dominant Token Selection, we first calculate the attention score as shown in Eq. 3,

$$S_h = \text{Softmax}\left(\frac{Q_h K_h^\top}{\sqrt{D_h}}\right), \qquad (3)$$

where S_h is the attention score of each head, and D_h is the head dimension, Q_h and K_h represent query and key, respectively. By averaging across the head dimension, we obtain an aggregated attention matrix $S_{avg} \in \mathbb{R}^{B \times SeqLen \times SeqLen}$, which reflects how each token attends to every other token. The above process is similar to that of vision encoders with a CLS token, as described in the main text.

To identify key visual tokens, we calculate the average attention each token receives from all others in the sequence. Specifically, we compute the average along dim=1 of S_{avg} to determine the degree to which each token is attended to by others, representing its importance. Tokens with higher average attention are considered more significant and are retained. We provide the pseudocode in Algorithm 3.

B. Additional Experiments

B.1. Image Understanding

B.1.1. Implementation Details.

Environments. We conduct the inference on a single NVIDIA A800-80G GPU, while the fine-tuning process is performed on 8 NVIDIA A800-80G GPUs. Furthermore, to demonstrate the efficiency and effectiveness of our VisionZip, the full training is conducted on 8 NVIDIA 3090-24G GPUs.

Parameters. For the VisionZip fine-tuning mode, we fine-tune only the cross-modality projector layer using a learning rate of $2e-5$, while keeping other components frozen. For the VisionZip training stage and inference mode, we follow the evaluation settings of the original model.

Algorithm 3 Pseudocode for Dominant Token Selection-NO CLS Token

```
# 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]

# no CLS token, use mean calculate received attention
attn_rec = attn.mean(dim=1).mean(dim=1) # (B, S)

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

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

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

Token Number. As shown in Table 8, for LLaVA-1.5 and Mini-Gemini, we present the number of dominant visual tokens and contextual visual tokens across three different configurations. Additionally, for LLaVA-NeXT, which contains 5 subfigures, we provide the number of dominant visual tokens and contextual visual tokens across three different configurations in Table 9.

B.1.2. Evaluation Benchmark

We conducted experiments on these widely used visual understanding benchmarks.

SEEDBench. SEEDBench [25] comprises 19,000 multiple-choice questions annotated by human assessors. The evaluation spans 12 distinct aspects, assessing the models' ability to recognize patterns in images and videos across both spatial and temporal dimensions.

MMMU. MMMU [62] evaluates multimodal models on complex tasks requiring college-level knowledge and reasoning. It includes 11.5K curated questions from exams, quizzes, and textbooks, spanning six disciplines: Art & Design, Business, Science, Health & Medicine, Humanities & Social Science, and Tech & Engineering. Covering 30 subjects and 183 subfields, these questions incorporate 30 image types like charts, diagrams, and chemical structures. MMMU challenges models with advanced perception and domain-specific reasoning, similar to expert-level.

MMVet. MMVet [60] defines six core vision-and-language (VL) capabilities: recognition, OCR, knowledge, language generation, spatial awareness, and math. These capabilities integrate to address a range of complex multimodal tasks. MM-Vet evaluates 16 specific integrations of these capabilities through quantitative assessments.

	Retain 64		Retain 128		Retain 192	
	Dominant	**Contextual**	**Dominant**	**Contextual**	**Dominant**	**Contextual**
LLaVA-1.5	54	10	108	20	162	30
Mini-Gemini	54	10	108	20	162	30

Table 8. Token number settings for VisionZip in LLaVA-1.5 [32] and Mini-Gemini [30]

	Retain 160		Retain 320		Retain 640	
	Dominant	**Contextual**	**Dominant**	**Contextual**	**Dominant**	**Contextual**
LLaVA NeXT	135	25	270	50	540	100

Table 9. Token number settings for VisionZip in LLaVA-NeXT [33]

LLaVA-Bench. LLaVA-Bench [32] collects a diverse set of 24 images paired with 60 questions, encompassing indoor and outdoor scenes, memes, paintings, sketches, and more. Each image is accompanied by a highly detailed, manually curated description and a carefully selected set of questions. This design also evaluates the model's robustness to various prompts. Additionally, LLaVA-Bench categorizes questions into three types: conversational (simple QA), detailed description, and complex reasoning.

VizWiz. VizWiz [14] comprises over 31,000 visual questions created by blind individuals, each capturing a photo using a mobile phone and recording a spoken question about it. Each visual question is paired with 10 crowdsourced answers. The images, taken by blind photographers, are often of lower quality, the questions are spoken and conversational, and some visual questions cannot be answered due to the nature of the content.

MMBench. MMBench [36] evaluates models through three hierarchical levels of abilities: L-1 with two core abilities (perception and reasoning), L-2 with six sub-abilities, and L-3 with 20 specific dimensions. This structure enables a detailed assessment of diverse capabilities.

ScienceQA. Spanning domains like natural, language, and social sciences, ScienceQA [38] organizes questions hierarchically into 26 topics, 127 categories, and 379 skills. This benchmark evaluates multimodal understanding, multi-step reasoning, and interpretability.

GQA. The GQA [19] benchmark evaluates visual scene understanding and reasoning using scene graphs, questions, and images. It includes spatial attributes and object features, with questions designed to test interpretation and reasoning.

POPE. POPE [28] evaluates Object Hallucination in models using binary questions on object presence in images. Metrics like Accuracy, Recall, Precision, and F1 Score measure hallucination levels across three sampling strategies, offering precise assessments.

MME. The MME [11] benchmark evaluates model performance across 14 subtasks targeting perceptual and cogni-

tive abilities. Using manually designed instruction-answer pairs, MME minimizes data leakage for fair assessment.

VQA-V2. VQA-V2 [13] tests visual perception using 265,016 images of real-world scenes and objects paired with open-ended questions. Each question includes 10 ground truth answers from human annotators for accurate evaluation.

TextVQA. TextVQA [45] evaluates a model's ability to interpret visual elements and embedded text in images through tasks requiring reasoning with textual information for accurate answers.

B.1.3. Additional Experiments for LLaVA-1.5

Effectiveness on 13B. In the main paper, we demonstrate the effectiveness of our model on 7B in Table 1, and we show the effectiveness of our model on 13B in this section. As shown in Table 10, we conduct our proposed VisionZip on 11 widely used evaluation benchmark. Due to the small size of LLaVA-Bench (LLaVA Wild Bench) and MMVeT, as well as the observation that their results can sometimes be unstable, we have excluded them from the average calculation in the last column. This decision was made despite our method demonstrating strong performance on both benchmarks. Instead, the average is calculated exclusively based on the 9 benchmarks. As shown in Table 10, we evaluate our method on three configurations of the vision token count (192, 128, and 64). The results show that even when retaining only 64 visual tokens, our method achieves 93.7% performance without requiring additional training time. In the efficient-tuning mode, this performance increases to 94.8%. Furthermore, when retaining 128 or 192 tokens, our method shows almost no performance loss in the 13B model.

Effectiveness on Training Stage. Our proposed method can also be applied during the training stage to reduce token length, thereby saving memory usage and training time. As shown in Table 11, we conduct experiments on three different vision token count configurations (192, 128, and 64). We apply our proposed VisionZip during the fine-tuning

Method	GQA	MMB	MME	POPE	SQA	VQA^{V2}	VQA^{Text}	MMMU	SEED-I	MMVet	LLaVA-B	Avg.
Upper Bound, 576 Tokens **(100%)**												
Vanilla (CVPR24)	63.2	67.7	1818	85.9	72.8	80.0	61.3	36.4	66.9	35.3	70.8	100%
	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	
Retain 192 Tokens (↓ **66.7%**)												
VisionZip	59.1	66.9	1754	85.1	73.5	78.1	59.5	36.4	65.2	37.5	77.5	**97.9%**
	93.5%	98.8%	96.5%	99.1%	101.0%	97.6%	97.1%	100%	97.5%	106.2%	109.5%	
VisionZip ‡	61.6	67.1	1790	84.5	72.7	78.6	59.9	36.4	66.1	37.7	73.9	**98.7%**
	97.5%	99.1%	98.5%	98.4%	99.9%	98.3%	97.7%	100%	98.8%	106.7%	104.3%	
Retain 128 Tokens (↓ **77.8%**)												
VisionZip	57.9	66.7	1743	85.2	74.0	76.8	58.7	36.1	63.8	37.5	70.8	**97.0%**
	91.6%	98.5%	95.9%	99.2%	101.6%	96.0%	95.8%	99.2%	95.4%	106.2%	100%	
VisionZip ‡	60.1	67.6	1736	83.8	73.0	77.6	59.2	35.4	64.9	38.3	72.3	**97.4%**
	95.1%	99.9%	95.5%	97.6%	100.2%	97.0%	96.6%	97.3%	97.0%	108.5%	102.1%	
Retain 64 Tokens (↓ **88.9%**)												
VisionZip	56.2	64.9	1676	76.0	74.4	73.7	57.4	36.4	60.4	33.9	70.3	**93.7%**
	88.9%	95.9%	92.2%	88.5%	102.2%	92.1%	93.3%	100%	90.3%	96.0%	99.3%	
VisionZip ‡	58.1	65.6	1671	81.6	72.3	75.2	58.5	35.3	61.4	36.7	68.7	**94.8%**
	91.9%	96.9%	91.9%	95.0%	99.3%	94.0%	95.4%	97.0%	91.8%	104.0%	97.0%	

Table 10. **Performance of VisionZip on LLaVA 1.5 13B.** 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. SEED-I represents SEED-IMG, which uses the metric from LMMs-Eval [64]. The Avg calculation process does not include the results from LLaVA-B and MMVet, as the benchmark is small and the results are not stable.

Method	GQA	MMB	MME	POPE	SQA	VQA^{V2}	VQA^{Text}	MMMU	SEED	MMVet	VizWiz	LLaVA-B	Avg.
Vanilla (CVPR24)	61.9	64.7	1862	85.9	69.5	78.5	58.2	36.3	58.6	31.1	50.0	66.8	100%
	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	
VisionZip 192 Tokens	61.5	67.4	1820	85.2	69.3	78.5	57.8	36.1	59.6	33	52.6	71.3	**100.6%**
	99.4%	104.2%	97.7%	99.2%	99.7%	100%	99.3%	99.4%	101.7%	106.1%	105.2	106.7%	
VisionZip 128 Tokens	60.0	66.6	1814	84.3	69.4	77.8	57.6	36.9	59.0	31.4	49.9	66.7	**99.6%**
	96.9%	102.9%	97.4%	98.1%	99.9%	99.1%	99.0%	101.7%	100.7%	101%	99.8%	99.9%	
VisionZip 64 Tokens	58.9	63.7	1785	84.1	69.3	76.0	57.1	36.2	55.8	29.9	46.8	63.5	**97.1%**
	95.2%	98.5%	95.9%	97.9%	99.7%	96.8%	98.1%	99.7%	95.2%	96.1%	93.6%	95.1%	

Table 11. **Using VisionZip train the LLaVA 1.5 7B.** 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. The Avg calculation process does not include the results from LLaVA-B and MMVet, as the benchmark is small and the results are not stable.

stage [32], with all hyperparameters, except for the batch size, following the vanilla training settings. All experiments are conducted on 8 Nvidia 3090 24G GPUs with a batch size of 4. To demonstrate the effectiveness of VisionZip in training mode, we evaluate it on 12 benchmarks and present the results. However, when calculating the average, we exclude LLaVA-Bench (LLaVA Wild Bench) and MMVet due to its small size and the observation that its results can be unstable, even though our method performs strongly on it.

The results show that even when the number of tokens is reduced to 128, 99.6% of the performance is retained. When retaining 192 tokens, performance even improves by 0.6%. We believe the reason is that reducing the redundancy of input visual tokens and providing only the more informative ones minimizes interference from less informative tokens. This allows the model to focus more on the informative tokens during training, enhancing visual understanding and leading to improved performance.

Method	GQA	MMB	MME	POPE	SQA	VQA^{V2}	VQA^{Text}	MMMU	SEED-I	Avg.
Upper Bound, 2880 Tokens (**100%**)										
Vanilla	64.2	67.9	1842	86.4	70.2	80.1	61.3	35.1	70.2	100%
	100%	100%	100%	100%	100%	100%	100%	100%	100%	
Retain 640 Tokens (↓ **77.8%**)										
VisionZip	61.3	66.3	1787	86.3	68.1	79.1	60.2	34.7	66.7	**97.5%**
	95.5%	97.6%	97.0%	99.9%	97.0%	98.8%	98.2%	98.9%	95.0%	
VisionZip ‡	62.4	65.9	1778	87.6	67.9	79.9	60.8	37.2	67.8	**98.9%**
	97.2%	97.1%	96.5%	101.4%	96.7%	99.8%	99.2%	106.0%	96.6%	
Retain 320 Tokens (↓ **88.9%**)										
VisionZip	59.3	63.1	1702	82.1	67.3	76.2	58.9	35.3	63.4	**94.5%**
	92.3%	92.9%	92.4%	95.0%	95.9%	95.1%	96.1%	100.5%	90.3%	
VisionZip ‡	61.0	64.4	1770	86.2	67.5	78.4	59.3	38.0	65.9	**97.6%**
	95.0%	94.8%	96.1%	99.8%	96.2%	97.9%	96.7%	108.3%	93.9%	
Retain 160 Tokens (↓ **94.4%**)										
VisionZip	55.5	60.1	1630	74.8	68.3	71.4	56.2	36.1	58.3	**91.5%**
	86.4%	88.5%	88.5%	86.6%	97.3%	89.1%	91.7%	102.8%	83.0%	
VisionZip ‡	58.2	63.9	1699	83.4	67.5	75.6	57.3	37.7	62.9	**95.0%**
	90.7%	94.1%	92.2%	96.5%	96.2%	94.4%	93.5%	107.4%	89.6%	

Table 12. **Performance of VisionZip on LLaVA NeXT 7B.** The vanilla number of visual tokens is 2880. 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. SEED-I represents SEED-IMG, which uses the metric from LMMs-Eval [64].

B.1.4. Additional Experiments for LLaVA-NeXT

In the main paper Table 2, we present the performance of VisionZip on LLaVA-NeXT across several evaluation benchmarks. The complete benchmark results are provided in Table 12. In this table, we only display the LLaVA NeXT 7B results for these stable benchmarks, and the results demonstrate that our proposed VisionZip consistently delivers strong performance.

To further demonstrate the effectiveness of our VisionZip, we present the results on the LLaVA-NeXT 13B model. As shown in Table 13, our method demonstrates excellent scalability. As the size of the LLM increases, the performance of VisionZip does not degrade. Our proposed VisionZip is highly adaptable to various sizes and types of LLMs, further highlighting the effectiveness of our approach. Notably, when retaining only 640 tokens, which eliminating 77.8% of the tokens, our method enables the 13B model to outperform the 7B model in training-free mode. Furthermore, the generation speed of our 13B model is faster, and we will provide detailed speed in the next section.

B.1.5. Additional Experiments for Mini-Gemini

In the main paper, Fig. 4 demonstrates that our method outperforms approaches like SparseVLM and FastV in terms of performance. Furthermore, as the number of retained to-

kens decreases, the performance advantage of our method becomes increasingly significant. In this section, we provide a detailed analysis of the results achieved by our method.

As shown in Table 14, the results indicate that after removing 88.9% of the tokens, our method can still retain over 90% of its performance in the training-free mode. Furthermore, with fine-tuning, its performance can reach up to 95%. When discarding 66.7% of the visual tokens, which is more than half, the performance remains virtually unaffected. These results further highlight the significant redundancy present in visual tokens.

B.1.6. Ablation Study

Impact of Fine-Tuning Dataset Compatibility We use VisionZip to efficiently fine-tune the cross-modality projector, addressing the gap caused by reduced visual tokens. Ensuring dataset compatibility with the original model is crucial for optimal performance. To evaluate this, we compare the effects of using 1/10 of the LLaVA 1.5 and LLaVA-NeXT datasets to fine-tune the LLaVA-NeXT model across three token count configurations (640, 320 and 160). As shown in Table 15, improving dataset compatibility results in minimal gains (less than 0.5%), with performance on some benchmarks even declining. These findings suggest that for efficient tuning to address token reduction, the ba-

Method	GQA	MMB	MME	POPE	SQA	VQA^{V2}	VQA^{Text}	MMMU	SEED-I	Avg.
Upper Bound, 2880 Tokens **(100%)**										
Vanilla 13B	65.4	70.0	1901	86.2	73.5	81.8	64.3	36.2	71.9	100%
	100%	100%	100%	100%	100%	100%	100%	100%	100%	
Vanilla 7B	64.2	67.9	1842	86.4	70.2	80.1	61.3	35.1	70.2	97.2%
	98.2%	96.3%	96.9%	100.2%	95.5%	97.9%	95.3%	97.0%	97.6%	
Retain 640 Tokens (↓ **77.8**%)										
VisionZip	63.0	68.6	1871	85.7	71.2	79.7	62.2	36.4	68.8	**97.5%**
	96.3%	98.0%	98.4%	99.4%	96.7%	96.9%	96.7%	100.5%	95.7%	
VisionZip ‡	63.7	66.6	1829	86.3	73.2	81.2	64.4	38.1	69.2	98.8%
	97.4%	95.1%	96.2%	100.1%	99.6%	99.3%	100.2%	105.2%	96.2%	
Retain 320 Tokens (↓ **88.9**%)										
VisionZip	60.7	67.2	1805	82.0	70.3	76.8	60.9	35.6	65.2	**94.7%**
	92.8%	96.0%	95.0%	95.1%	95.6%	93.9%	94.7%	98.3%	90.7%	
VisionZip ‡	62.5	66.9	1861	85.7	72.7	80.0	63.2	36.9	67.9	97.8%
	95.6%	95.6%	97.9%	99.4%	98.9%	97.8%	98.3%	101.9%	94.4%	
Retain 160 Tokens (↓ **94.4**%)										
VisionZip	57.8	64.9	1739	76.6	69.3	72.4	58.4	37.0	61.1	**91.3%**
	88.4%	92.7%	91.5%	88.9%	94.3%	88.5%	90.8%	102.2%	84.8%	
VisionZip ‡	59.7	65.3	1766	84.0	72.0	77.6	60.8	36.0	64.4	94.6%
	91.3%	93.3%	92.9%	97.4%	98.0%	94.9%	94.6%	99.4%	89.6%	

Table 13. **Performance of VisionZip on LLaVA NeXT 13B.** The vanilla number of visual tokens is 2880. 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. SEED-I represents SEED-IMG, which uses the metric from LMMs-Eval [64].

sic 1/10 LLaVA 1.5 dataset is sufficient. The results further demonstrate that the performance gains of VisionZip‡ in Table 1 and Table 2 of the main text are not attributable to additional knowledge acquired through continued training. Instead, these improvements arise from adaptation to the sudden reduction in tokens, which helps bridge the gap between the visual and LLM spaces. This finding aligns with our motivation outlined in Sec. 2.4.

B.2. Video Understanding

B.2.1. Evaluation Benchmark

TGIF-QA. TGIF-QA [20] extends ImageQA to videos with 165,000 question-answer pairs based on GIFs. It includes three VideoQA tasks—repetition count, repeating action, and state transition—requiring spatio-temporal reasoning, plus frame QA tasks answerable from single frames.

MSVD-QA. MSVD-QA [54], based on the MSVD dataset, features 1,970 video clips and 50.5K question-answer pairs. Covering diverse topics, it supports video question answering and captioning with open-ended questions in five categories: what, who, how, when, and where.

MSRVTT-QA. MSRVTT-QA [54] includes 10,000 video clips and 243,000 question-answer pairs, emphasizing

video understanding and reasoning. Questions, categorized into what, who, how, when, and where, require models to process visual and temporal information.

ActivityNet-QA. ActivityNet-QA [61] consists 58,000 human-annotated question-answer pairs from 5,800 ActivityNet videos. Covering motion, spatial, and temporal relationships, it evaluates VideoQA models on long-term spatio-temporal reasoning.

B.2.2. Future Direction

With the development of LLMs and VLMs, video understanding has become a popular research direction. Whether the goal is for VLMs to comprehend longer videos or to achieve precise localization within videos, enabling the input of more frames within limited memory is both important and critical.

However, existing methods process a single frame into at least 256 tokens, which hinders the ability to input more frames. With our approach, VisionZip, the number of video tokens can be reduced by 5-10 times before being input into the LLM. This reduction allows the model to process 5-10 times more frames within the same memory constraints. For example, if a model could originally handle only 1 hour of

Method	GQA	MMB	MME	POPE	SQA	VQA^{V2}	VQA^{Text}	MMMU	SEED-I	Avg.
Upper Bound, 576 Tokens (**100%**)										
Vanilla 7B	62.4	69.3	1841	85.8	70.7	80.4	65.2	36.1	69.7	100%
	100%	100%	100%	100%	100%	100%	100%	100%	100%	
Retain 192 Tokens (↓ **66.7**%)										
VisionZip	60.3	68.9	1846	82.3	70.1	79.1	63.4	36.1	67.5	**98.2%**
	96.6%	99.4%	100.2%	95.9%	99.2%	98.4%	97.2%	100%	96.8%	
VisionZip ‡	61.6	67.2	1804	85.5	70.2	78.9	63.6	36.1	67.0	**98.3%**
	98.7%	97.0%	98.0%	99.7%	99.3%	98.1%	97.5%	100%	96.1%	
Retain 128 Tokens (↓ **77.8**%)										
VisionZip	58.7	68.1	1841	78.5	70.0	77.5	61.3	34.8	65.6	**96.0%**
	94.1%	98.3%	100%	91.5%	99.0%	96.4%	94.0%	96.4%	94.1%	
VisionZip ‡	60.0	67.0	1810	83.2	70.1	78.3	61.6	34.8	65.9	**96.7%**
	96.2%	96.7%	98.3%	97.0%	99.2%	97.4%	94.5%	96.4%	94.5%	
Retain 64 Tokens (↓ **88.9**%)										
VisionZip	55.8	65.9	1737	69.6	70.7	73.9	59.1	35.6	61.7	**92.2%**
	89.4%	95.1%	94.4%	81.4%	100%	91.9%	90.6%	98.6%	88.5%	
VisionZip ‡	57.7	66.3	1779	80.0	71.0	75.9	60.1	36.2	62.6	**95.0%**
	92.5%	95.7%	96.6%	93.2%	100.4%	94.4%	92.2%	100.3%	89.8%	

Table 14. **Performance of VisionZip on mini-Gemini 7B.** 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. SEED-I represents SEED-IMG, which uses the metric from LMMs-Eval [64].

Dataset	GQA	MMB	MME	SQA	VQA^{V2}	VQA^{Text}	MMMU	Avg.
Retain 640 Tokens (↓ **77.8**%)								
LLaVA-1.5	62.4	65.9	1778	67.9	79.9	60.8	37.2	98.9%
LLaVA-NeXT	63.0	66.8	1738	68.4	80.1	61.2	38.8	99.3%
Retain 320 Tokens (↓ **88.9**%)								
LLaVA-1.5	61.0	64.4	1770	67.5	78.4	59.3	38.0	97.6%
LLaVA-NeXT	61.6	64.7	1771	67.5	78.8	60.1	36.3	97.3%
Retain 160 Tokens (↓ **94.4**%)								
LLaVA-1.5	58.2	63.9	1699	67.5	75.6	57.3	37.7	95.2%
LLaVA-NeXT	58.4	63.2	1763	68.0	76.0	58.2	36.9	95.7%

Table 15. **Impact of Fine-Tuning Dataset Compatibility.** The first column indicates which dataset was used to sample 1/10 of the data for fine-tuning the multimodality projector.

video, VisionZip enables it to process 5-10 hours of video, significantly enhancing the application value of VLMs in video understanding.

As shown in Fig. 8, we select 3-minute video clips from Zootopia, a well-known cartoon, and ask the model to describe it. The results show that VideoLLaVA tends to describe a single frame in detail, lacking an overall understanding of the video, as it can only encode an 8-frame video. In contrast, our VisionZip can encode 10× more video frames without increasing the token count, signifi-

cantly enhancing the model's ability to understand longer videos.
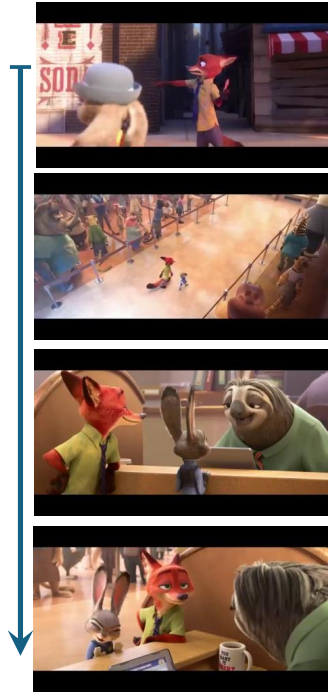
B.3. Efficiency Analysis

In this section, we provide additional results highlighting the efficiency gains brought by VisionZip.

CUDA Memory Save. We conduct experiments on the LLaVA-NeXT 13B model, retaining only 320 visual tokens. Additionally, to better illustrate the memory consumption changes introduced by this process, we simultaneously present the performance variations alongside the CUDA memory changes in ScienceQA. The result aligns with Table 6 in the main paper. As shown in Table 16, the third row demonstrates that using VisionZip can reduce CUDA memory consumption by more than 20%. Additionally, employing 8-bit and 4-bit quantization further reduces memory usage. Moreover, our method integrates seamlessly with quantization techniques, and the performance of the quantized model is comparable to the original results.

Training Time Save. Our proposed VisionZip can also reduce training time. We conducted an experiment on LLaVA-NeXT 7B, retaining 640 visual tokens. As shown in Table 17, using VisionZip during the training stage significantly reduces training time by 2× and achieves better performance compared to applying VisionZip only during

Please describe this video.

*Video-LLaVA encodes only **8 frames**, which results in a lack of detailed information.*

In the video, we see a group of animals gathered around a table, they seem to be discussing something important and the scene is strange.

VideoLLaVA

*VisionZip encodes **120 frames**, reducing redundancy while capturing more detailed information.*

The video seems to be a cartoon or animated movie scene. The video features a red fox and a rabbit character running through the streets and entering a house. Inside, the fox and rabbit walk up to a counter where a sloth is seen sitting at a table with a laptop.

VisionZip

3-minute clips from Zootopia

Figure 8. **Advantage of VisionZip in video understanding task.** With the same visual token length, using VisionZip allows encoding more frames, significantly enhancing the model's capacity to understand longer video sequences and capture more detailed information.

Method	Memory	GQA	MMB	MME	POPE	SQA	VQA^{V2}	VQA^{Text}	MMMU	SEED-I	Avg.
Upper Bound, 2880 Tokens (**100%**)											
Vanilla 13B	36721Mb	65.4	70.0	1901	86.2	73.5	81.8	64.3	36.2	71.9	100%
		100%	100%	100%	100%	100%	100%	100%	100%	100%	
Vanilla 7B	18952Mb	64.2	67.9	1842	86.4	70.2	80.1	61.3	35.1	70.2	97.2%
		98.2%	96.3%	96.9%	100.2%	95.5%	97.9%	95.3%	97.0%	97.6%	
Retain 320 Tokens (↓ **88.9%**)											
VisionZip	28810Mb	60.7	67.2	1805	82.0	70.3	76.8	60.9	35.6	65.2	**94.7%**
		92.8%	96.0%	95.0%	95.1%	95.6%	93.9%	94.7%	98.3%	90.7%	
VisionZip-8bit	16632Mb	60.6	67.1	1798	81.4	70.8	76.8	60.5	37.0	65.4	**95.0%**
		92.7%	95.9%	94.6%	94.4%	96.3%	93.9%	94.1%	102.2%	91.0%	
VisionZip-4bit	10176Mb	60.3	65.1	1773	82.1	70.3	76.6	60.0	36.1	65.1	**94.0%**
		92.2%	93.0%	93.3%	95.2%	95.6%	93.6%	93.3%	99.7%	90.5%	

Table 16. **Performance and Memory of VisionZip on LLaVA NeXT 13B with the Quantization.** The vanilla number of visual tokens is 2880. 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. SEED-I represents SEED-IMG, which uses the metric from LMMs-Eval [64]. The memory refers to the practical CUDA memory usage on a single Nvidia A800 GPU for SQA.

the inference stage.

Inference Time Save. To demonstrate the relationship be-

tween the number of remaining tokens and inference time, we conduct experiments on LLaVA-NeXT 13B. We config-

Method	Time	Memory	GQA	MMB	MME	POPE	SQA	VQA^{V2}	VQA^{Text}	MMMU	SEED-I	Avg.
Upper Bound, 2880 Tokens (**100%**)												
Vanilla 7B	33.8h	63558Mb	64.2	67.9	1842	86.4	70.2	80.1	61.3	35.1	70.2	100%
			100%	100%	100%	100%	100%	100%	100%	100%	100%	
Retain 640 Tokens (↓ **77.8%**)												
VisionZip-Inference			61.3	66.3	1787	86.3	68.1	79.1	60.2	34.7	66.7	**97.5%**
			95.5%	97.6%	97.0%	99.9%	97.0%	98.8%	98.2%	98.9%	95.0%	
VisionZip-Train	15.9h	35326Mb	62.5	67.1	1728	86.0	70.2	80.6	64.1	35.1	67.8	99.0%
			97.4%	98.8%	93.8%	99.5%	100%	100.6%	104.6%	100%	96.6%	

Table 17. **Performance and Training Time of VisionZip on LLaVA NeXT 7B.** The vanilla number of visual tokens is 2880. 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. SEED-I represents SEED-IMG, which uses the metric from LMMs-Eval [64]. The time refers to the practical Training time usage on 8 Nvidia A800 GPUs for training.

Method	Count	Prefilling	Total	GQA	MMB	MME	POPE	SQA	VQA^{V2}	VQA^{Text}	MMMU	SEED-I	Avg.
Vanilla 13B	2880	129.4ms	2506s	65.4	70.0	1901	86.2	73.5	81.8	64.3	36.2	71.9	100%
				100%	100%	100%	100%	100%	100%	100%	100%	100%	
Vanilla 7B	2880	54.2ms	1598s	64.2	67.9	1842	86.4	70.2	80.1	61.3	35.1	70.2	97.2%
				98.2%	96.3%	96.9%	100.2%	95.5%	97.9%	95.3%	97.0%	97.6%	
VisionZip 13B	640	48.2ms	1219s	63.0	68.6	1871	85.7	71.2	79.7	62.2	36.4	68.8	97.5%
				96.3%	98.0%	98.4%	99.4%	96.7%	96.9%	96.7%	100.5%	95.7%	
VisionZip 13B	320	30.3ms	995s	60.7	67.2	1805	82.0	70.3	76.8	60.9	35.6	65.2	94.7%
				92.8%	96.0%	95.0%	95.1%	95.6%	93.9%	94.7%	98.3%	90.7%	
VisionZip 13B	160	23.9ms	888s	57.8	64.9	1739	76.6	69.3	72.4	58.4	37.0	61.1	91.3%
				88.4%	92.7%	91.5%	88.9%	94.3%	88.5%	90.8%	102.2%	84.8%	

Table 18. **Performance of VisionZip on LLaVA NeXT 13B.** The vanilla number of visual tokens is 2880. 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. "Prefilling" represents the prefilling time, and "Total" represents the actual testing time of the model on the TextVQA benchmark.

ured three vision token counts: 640, 320, and 160, respectively. We recorded the prefilling time and the actual testing time on the benchmark. Specifically, we use the TextVQA dataset to conduct the time measurements. As shown in Table 18, by using VisionZip to retain 640 tokens, the 13B model achieves faster inference than the 7B model while maintaining superior performance.

C. Related Work

Vision-Language Models. Building on the success of LLMs [1, 2, 7, 9, 24, 27, 42, 49, 68], VLMs have made significant advancements [18, 23, 30, 32–34, 48, 50, 57, 66, 69]. Popular VLM models, such as LLaVA [32] and mini-Gemini [30], process visual tokens through a projector before inputting them into the LLM as a sequence. However, real-world images are typically high-resolution and require a large number of tokens. For example, LLaVA-NeXT processes 672×672 images into more than 2,000 tokens [33]. Moreover, handling videos or multiple images significantly

increases token requirements [17, 29, 31, 39, 46, 47]. . Hence, it's essential to discuss more efficient ways to extract information from visual tokens, rather than merely increasing their length.

Efficient Large Language Models. In the field of large language models (LLMs), various strategies have been developed to reduce tokens, thereby accelerating inference and optimizing key-value (KV) cache compression [15]. For example, StreamingLLM [52] decreases the KV cache size by retaining only the attention sinks and the most recent tokens. FastGen [12] introduces an adaptive method for managing the KV cache, dynamically optimizing memory usage by adjusting retention strategies based on the behavior of attention heads. Similarly, the Heavy-Hitter Oracle (H2O) [67] employs a scoring mechanism based on cumulative attention to selectively prune key-value pairs during the generation process. These methods aim to reduce token redundancy and enhance the efficiency of inference operations in LLMs.

Efficient Vision Language Models. Recently, some studies [6, 16, 44, 51, 53, 65] have also recognized the redundancy in visual tokens and proposed various methods to address it. And most of these works identify redundancy based on the relatively low attention that LLM text tokens assign to visual tokens. Furthermore, these studies primarily achieve token reduction or KV cache compression by leveraging attention mechanisms between text and visual tokens during the LLM forward process. In contrast to these works, we find that the visual tokens generated by popular vision encoders exhibit significant redundancy. Our approach removes this redundancy before the tokens are input into the LLM. Additionally, in Sec. 4 of the main paper, we provide a thorough comparison and analysis of our method against these text-relevant approaches.

D. Visualization

D.1. Visualization of Redundancy

To further show the redundancy in popular vision encoders, we include additional examples from the COCO train2017 dataset. This dataset is a key component of the LLaVA 1.5 fine-tuning dataset and an essential part of many vision datasets. As shown in Fig. 9 Fig. 10 and Fig. 11, the visualization results indicate that only a few tokens receive high attention and contain substantial amounts of information, while most visual tokens receive minimal attention and contain limited information. This visualization highlights the significant redundancy present in the visual tokens.

D.2. Visualization of Attention Distribution Change

In Sec. 4 of the main text, we discuss the reasons behind the redundancy in visual tokens. In this section, we present a comprehensive analysis of the changes in attention within the CLIP model. As shown in Fig. 12 and Fig. 13 attention in the early layers is broadly distributed across the image. However, by the middle layers, it rapidly converges onto a few tokens. In the deeper layers, attention and information become concentrated on a small set of dominant tokens, reaching peak concentration by the 23rd layer, which is used for visual token extraction for the LLM. Besides, in the final layer, attention is more dispersed as these tokens align with the CLIP text branch via contrastive loss, potentially limiting their ability to represent the original image.

D.3. Visualization of Feature Misalignment

In Fig. 6 of the main text, we show the phenomenon of feature misalignment. To further demonstrate that this phenomenon is widespread, we observe it across additional COCO images.

As shown in Fig. 14, in the first three columns, we select a token (red point) from the main subject of the figure and illustrate the attention to that token, and the last column shows that the attention score for the whole figure. The results show that the attention of the selected token does not focus on semantically relevant tokens but instead on dominant tokens, highlighting the phenomenon of feature misalignment. Hence, when text-relevant methods like SparseVLM [65] select tokens based on semantic relationships, they can identify semantically relevant tokens. However, these tokens contain less information compared to the dominant tokens, which aggregate information from the entire image.

In addition, to improve visualization and analysis, we developed a Gradio demo, as shown in Fig. 15. The corresponding code is provided on the GitHub page.
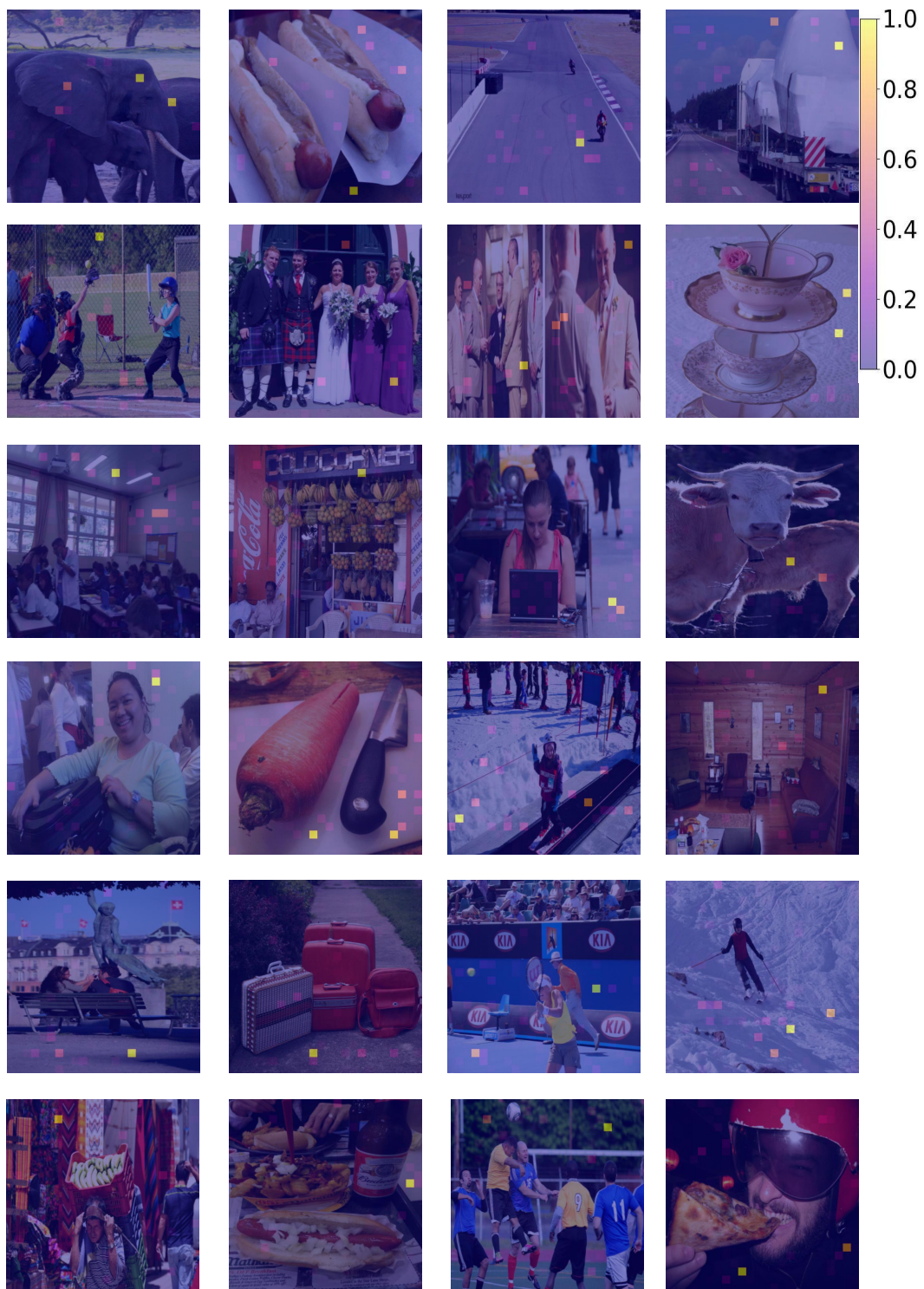
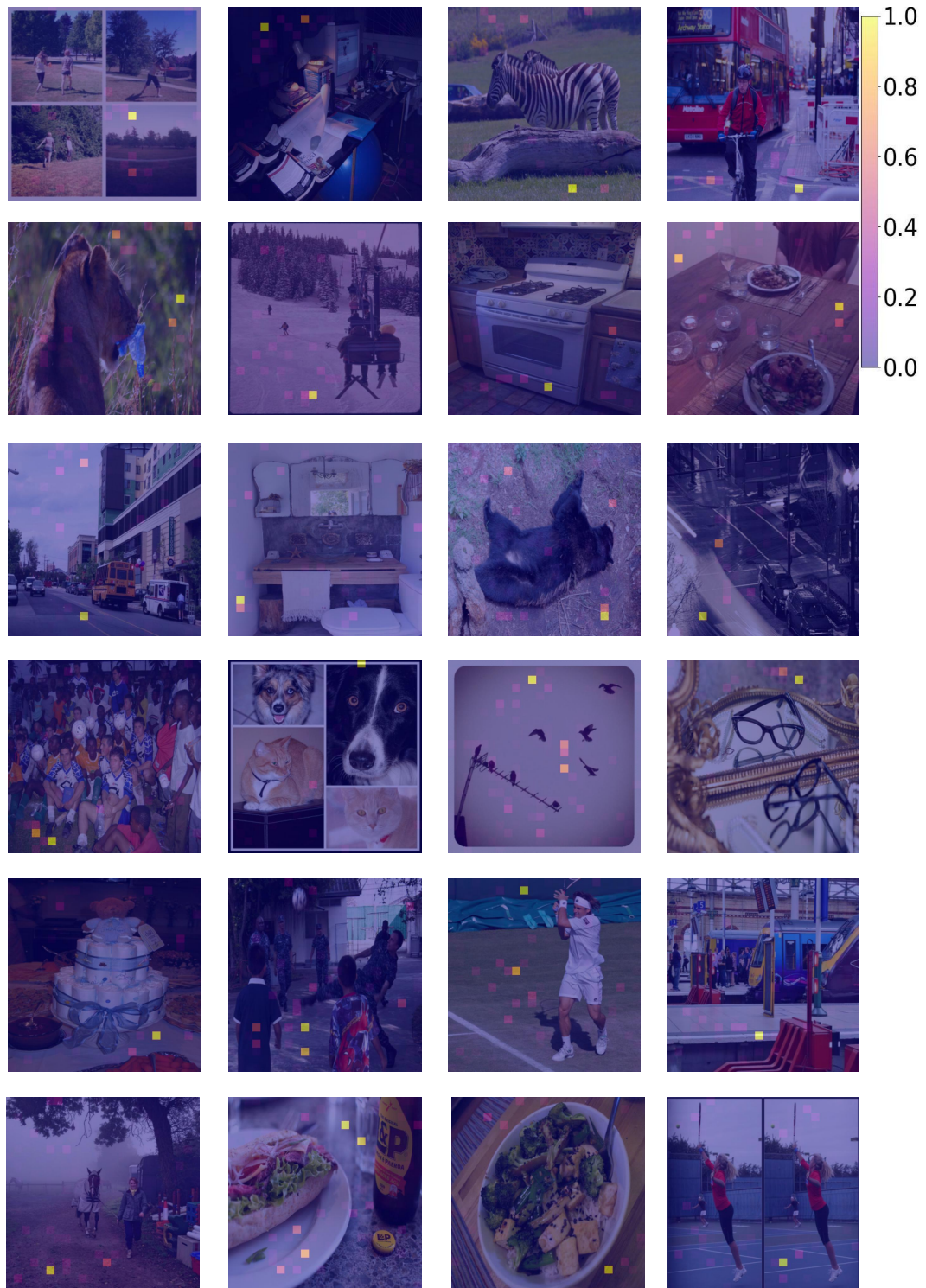Figure 9. Visualization of Redundancy in the CLIP Model

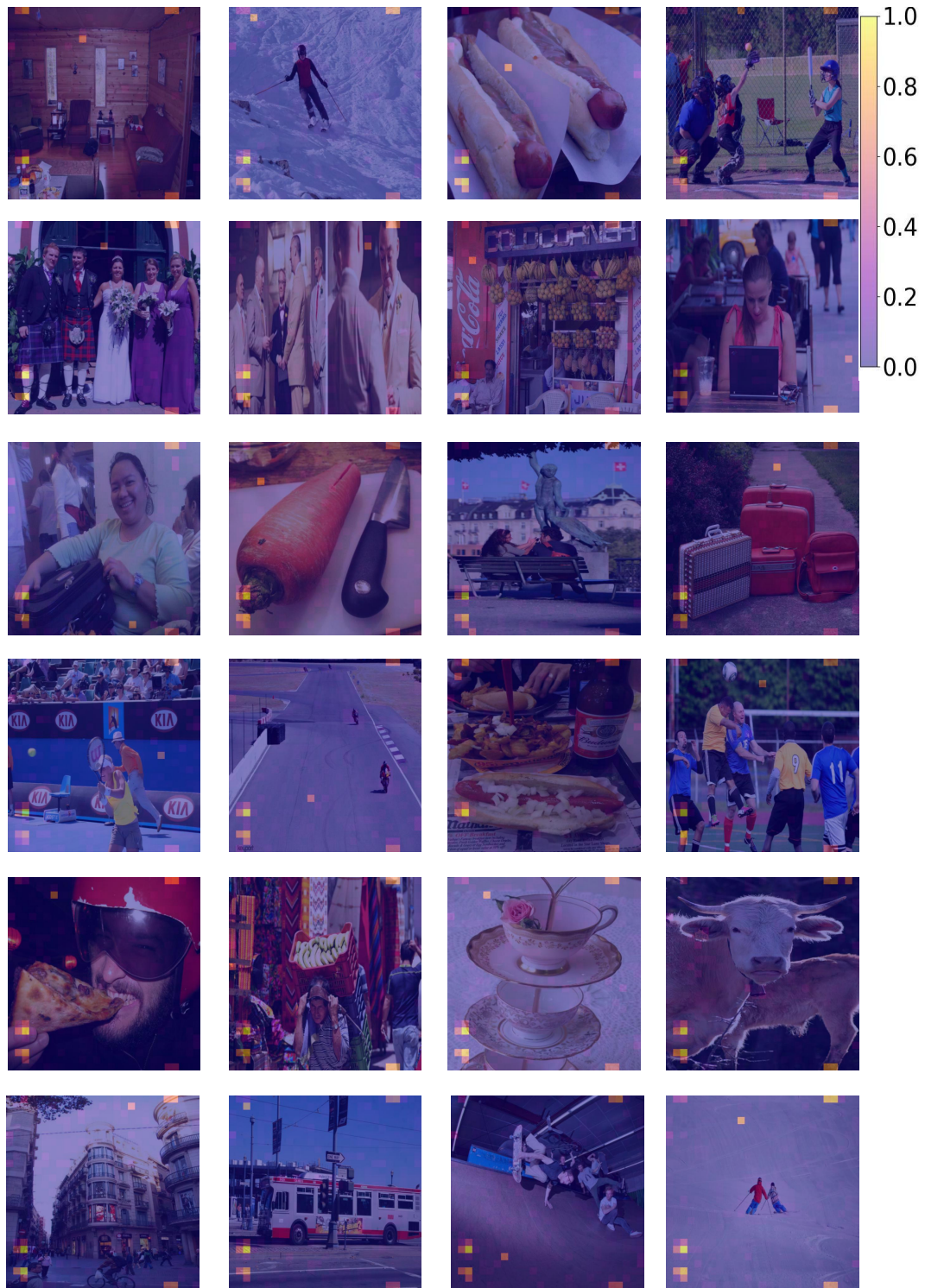Figure 10. Visualization of Redundancy in the CLIP Model

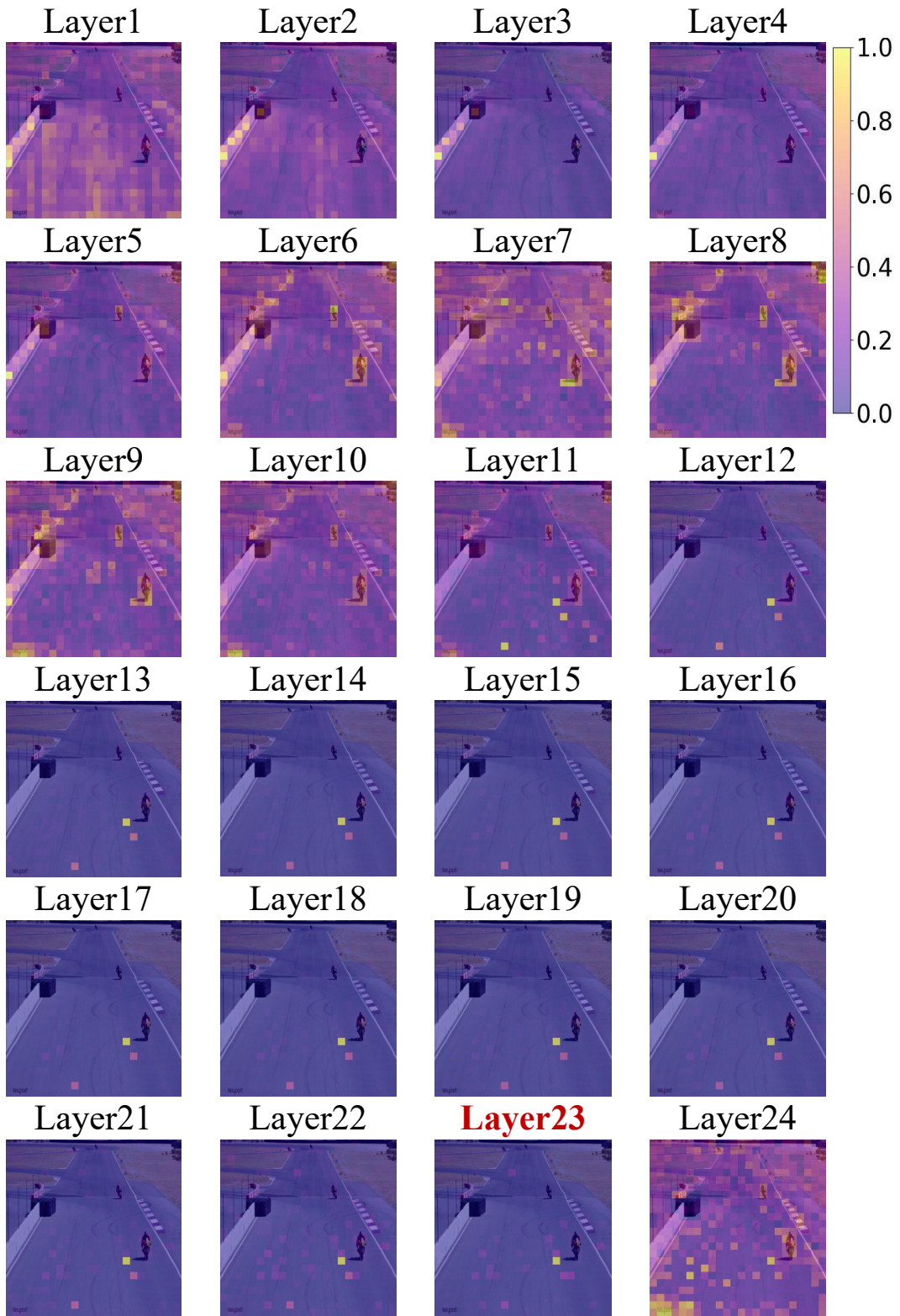Figure 11. Visualization of Redundancy in the SigLIP Model

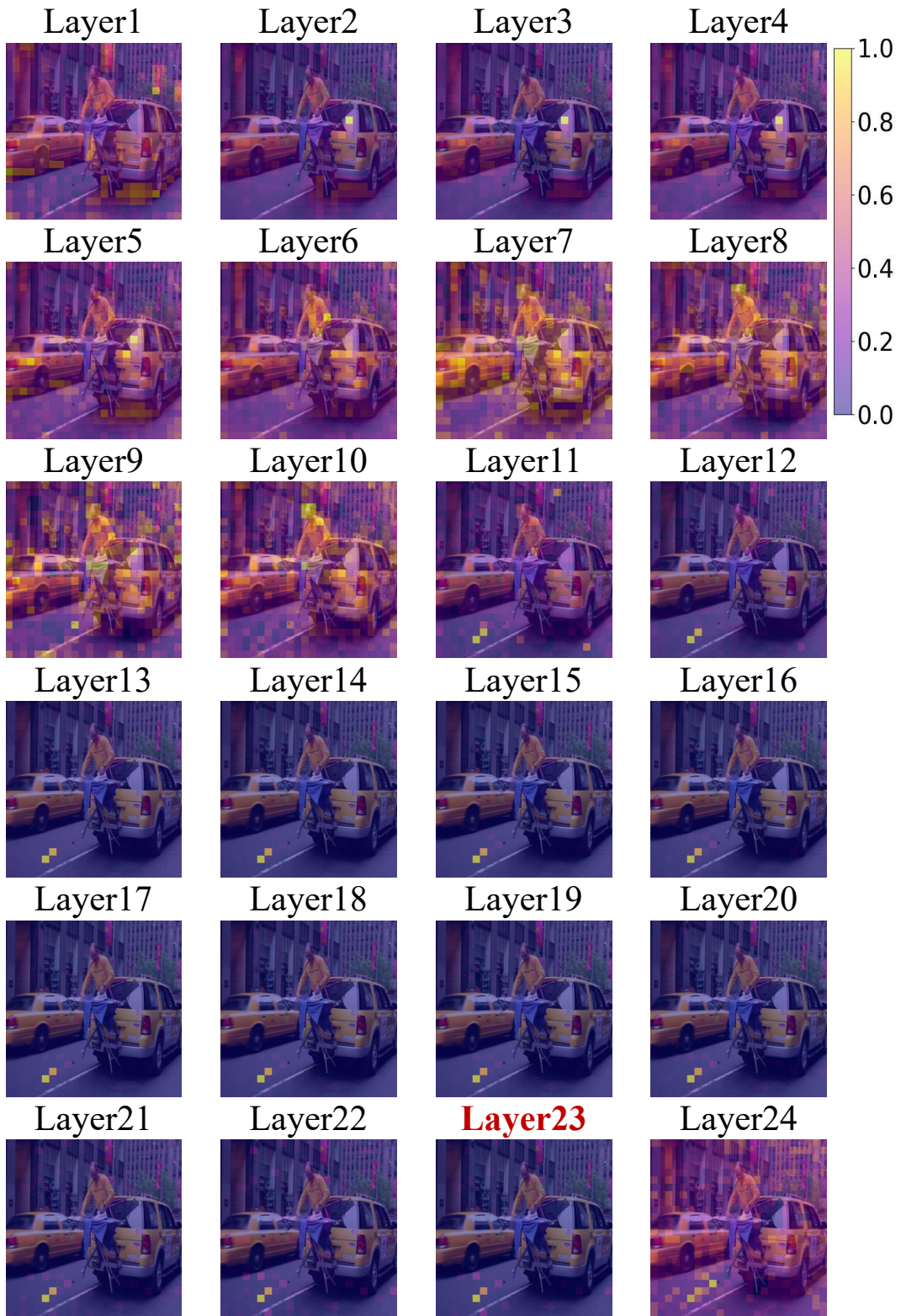Figure 12. Visualization of Attention Distribution Change

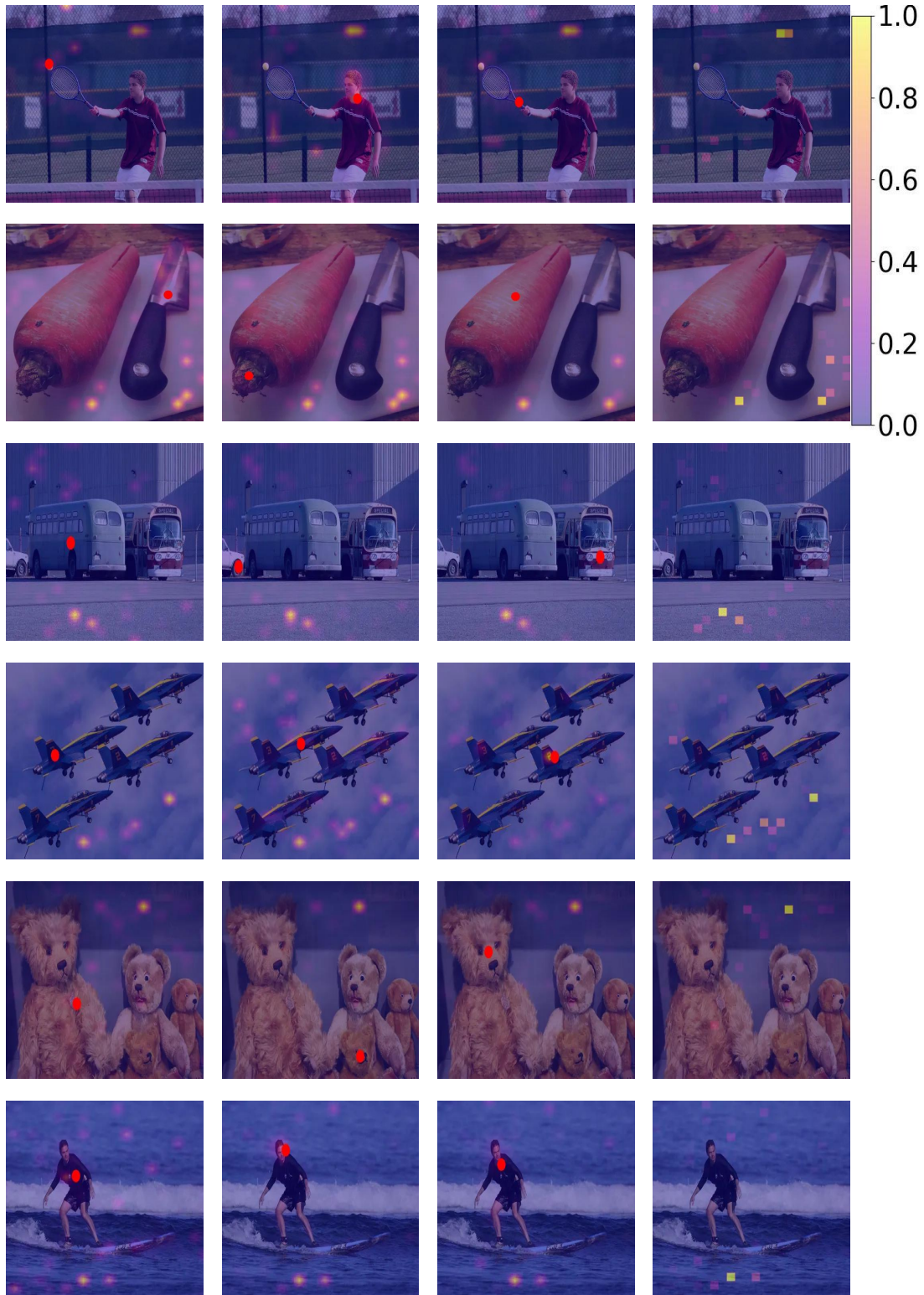Figure 13. Visualization of Attention Distribution Change

Figure 14. **Visualization of Feature Misalignment.** The red point represents the selected token, while the heatmaps in the first three columns illustrate the attention relationships to the selected token. The last column displays the attention map for the entire image. The results shows that the attention of the selected token does not focus on semantically similar tokens but instead on dominant tokens, highlighting the phenomenon of feature misalignment.

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

Redundancy and Feature Misalignment Visualizer

This tool enables the visualization of attention mechanisms in CLIP by analyzing redundancy and feature misalignment in token attention.
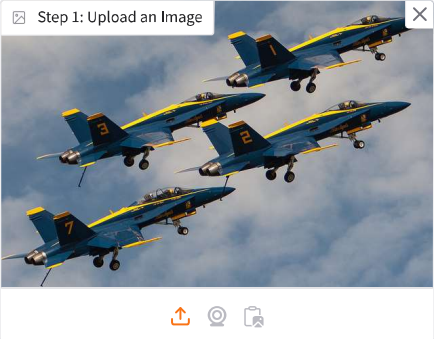
Features

○ **Attention to the Selected Token**: Displays the attention heatmap of the selected token across all patches.

○ **Patch Attention Heatmap**: Visualizes the relationships and redundancy between visual patches.

Insights

○ The **first heatmap** shows that the selected token's attention focuses on dominant tokens rather than semantically related tokens.

○ The **second heatmap** shows attention concentrated on a few tokens, emphasizing the redundancy in visual tokens.

Instructions

1. **Upload** an image using the left panel.

2. **Click** on a specific point in the image to analyze.

3. **View** the generated heatmaps below for insights.

Figure 15. Gradio demo to analysis the visual redundancy and the feature misalignment