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

Supplementary Material

Contents

A. Further Discussion

A.1. Comparison with Text-relevant Efficient VLM

We observe that most recent Efficient VLMs [6, 16, 55, 66] 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 outside 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 [42], OpenCLIP, and LanguageBind [71], 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,

$$\boldsymbol{S}_h = \text{Softmax}\left(\frac{\boldsymbol{Q}_h \boldsymbol{K}_h^\top}{\sqrt{D_h}}\right), \tag{3}$$

where $\boldsymbol{S}_h$ is the attention score of each head, and D_h is the head dimension, $\boldsymbol{Q}_h$ and $\boldsymbol{K}_h$ represent query and key, respectively. By averaging across the head dimension, we obtain an aggregated attention matrix $\boldsymbol{S}_{avg} \in \mathbb{R}^{B \times SeqLen \times SeqLen}$, 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 $\boldsymbol{S}_{avg}$ 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.

A.3. Additional Advantage of the VisionZip

Easy to deploy. 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 8, our method is compatible with existing quantization techniques, maintaining performance while minimizing

	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 6. Token number settings for VisionZip in LLaVA-1.5 [33] and Mini-Gemini [31]

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

Table 7. Token number settings for VisionZip in LLaVA-NeXT [34]

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.

memory usage. Furthermore, our method enables the 13B model to be faster and perform better than the 7B model. As shown in Table 9, 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.

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.

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

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.

Token Number. As shown in Table 6, 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 7.

B.1.2. Evaluation Benchmark

We conducted experiments on these widely used visual understanding benchmarks.

SEEDBench. SEEDBench [26] 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 [63] 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

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 [65]. 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.

Model	InfoVQA	DocVQA
Retain 192 Tokens (↓ **66.7%**)		
LLaVA 1.5	25.7	28.1
	100%	100%
VisionZip	25.0	25.8
	97.3%	91.8%

Table 11. **Effectiveness of VisionZip on OCR Benchmarks.**

& 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 [61] 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.

LLaVA-Bench. LLaVA-Bench [33] 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 [37] 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 [39] organizes questions hierarchically into 26 topics, 127 categories, and 379 skills. This benchmark evaluates multimodal understanding, multi-step reasoning, and interpretability.

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 12. **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.

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 13. **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 [65].

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 [29] 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 cognitive 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 [47] evaluates a model's ability to interpret visual elements and embedded text in images

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 14. **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 [65].

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 OCR Benchmarks. To demonstrate the effectiveness of our VisionZip on OCR-heavy benchmarks, we select the widely used InfoVQA and DocVQA datasets. As shown in Table 11, the results indicate that VisionZip does not experience significant performance degradation under OCR-heavy settings.

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 12, we conduct experiments on three different vision token count configurations (192, 128, and 64). We apply our proposed VisionZip during the fine-tuning stage [33], 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%.

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 15. **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 [65].

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.

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 13. 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 14, 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 tokens 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 15, 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

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 16. **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.

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 16, 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 basic $1/10$ 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 [56], 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 [56] 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 [62] 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 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 $10\times$ more video frames without increasing the token count, significantly 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 8 in the main paper. As shown in Table 17, 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 18, using VisionZip during the training stage significantly reduces training time by $2\times$ and achieves better performance compared to applying VisionZip only during the inference stage.

Inference Time Save. To demonstrate the relationship between the number of remaining tokens and inference time,

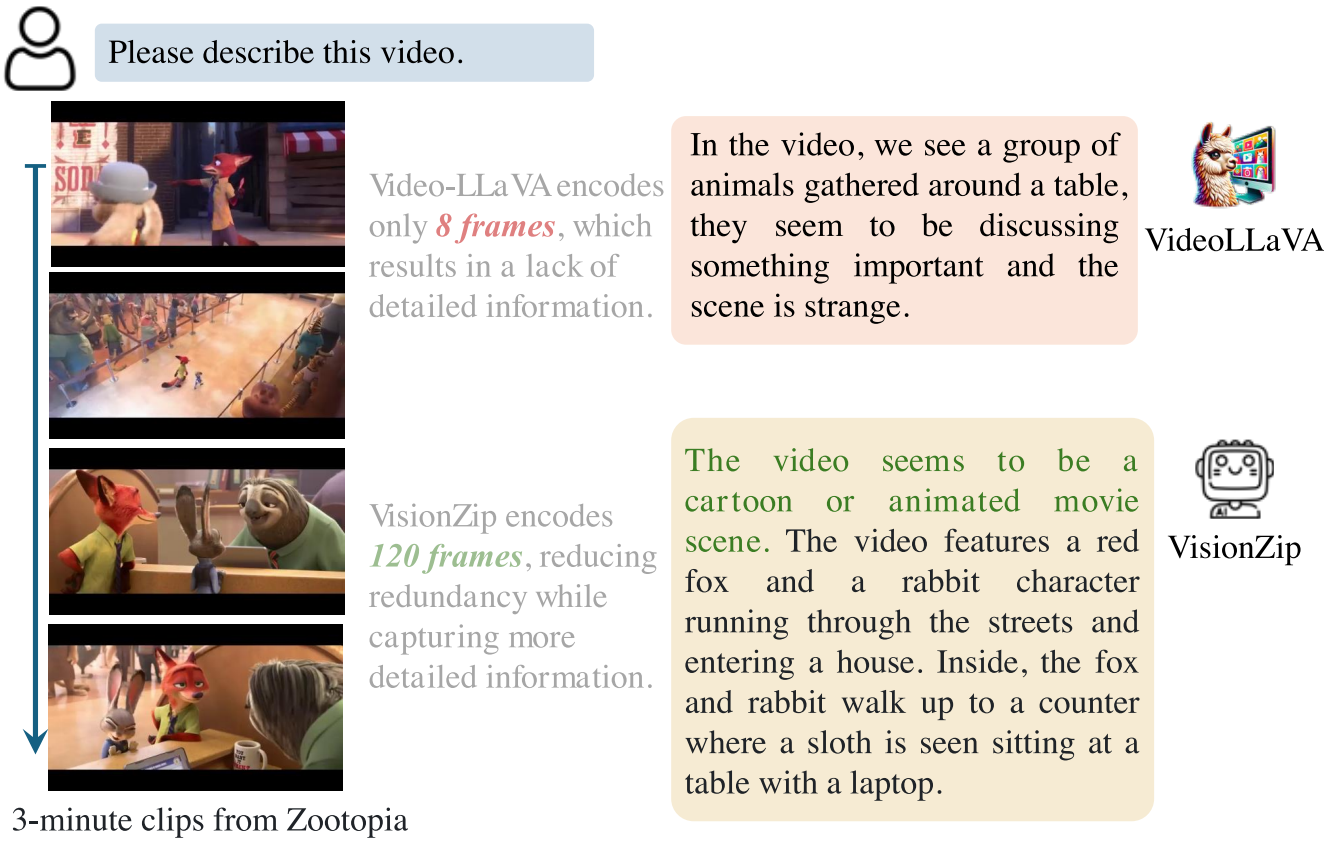


Figure 8. **Advantage of VisionZip in video understanding task.** With the same visual token length, using VisionZip allows encoding more frames, significantly enhancing the model's capacity to understand longer video sequences and capture more detailed information.

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 17. **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 [65]. The memory refers to the practical CUDA memory usage on a single Nvidia A800 GPU for SQA.

we conduct experiments on LLaVA-NeXT 13B. We configured 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 19, by using VisionZip to retain 640 tokens, the 13B

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 18. **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 [65]. 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 19. **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.

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, 25, 28, 43, 51, 69], VLMs have made significant advancements [18, 24, 31, 33–35, 50, 52, 58, 67, 70]. Popular VLM models, such as LLaVA [33] and mini-Gemini [31], 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 [34]. Moreover, handling videos or multiple images significantly increases token requirements [17, 30, 32, 40, 48, 49]. . 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 [54] 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) [68] 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 [16, 46, 53] have also recognized the redundancy in visual tokens and proposed various methods to address it. Specifically, EVLGen [21] utilizes the token merging strategy, while LLaVA-PruMerge [44] employs a clustering method to reduce the tokens. Additionally, several recent works [6, 55, 66] 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 [66] 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.

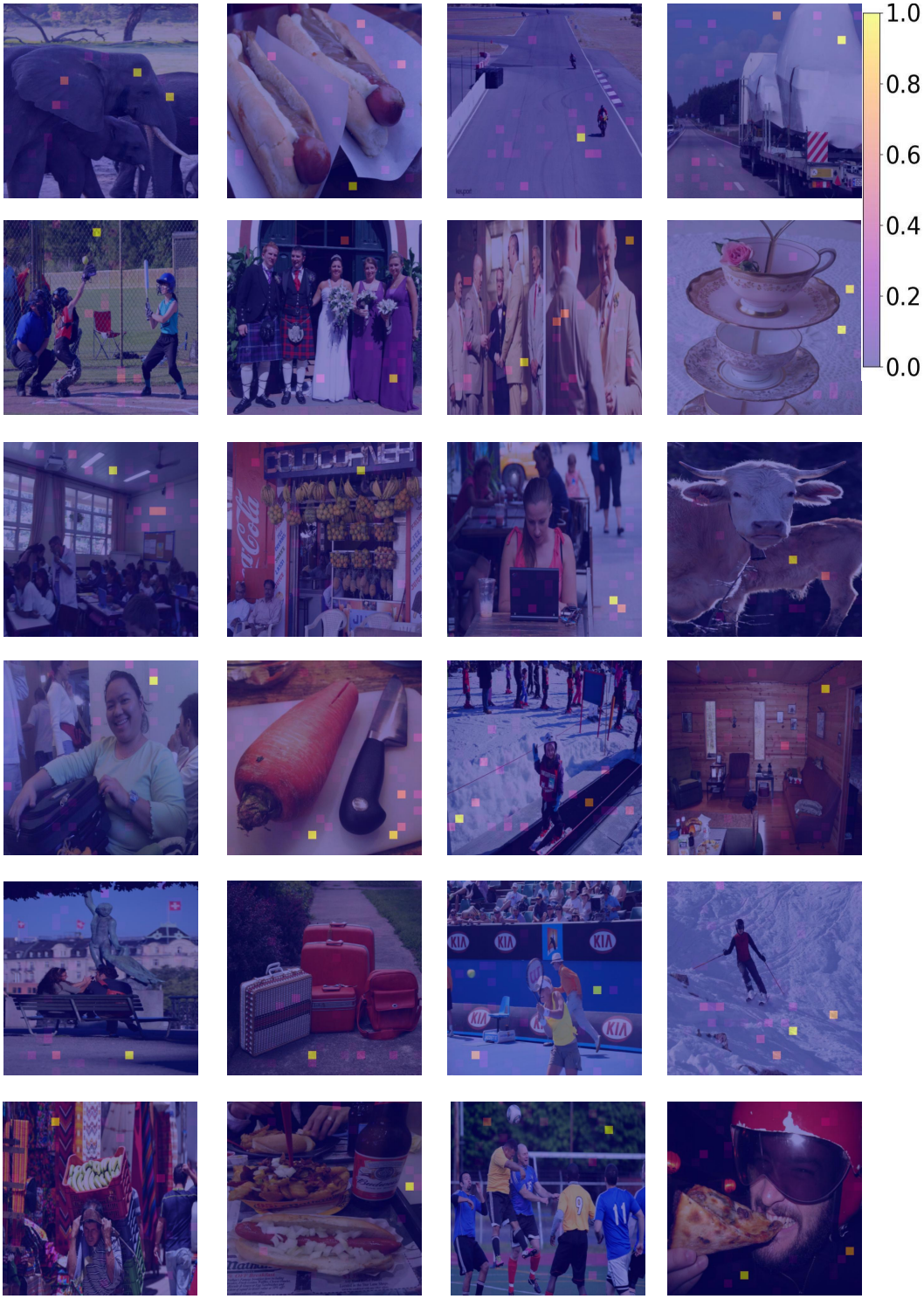


Figure 9. Visualization of Redundancy in the CLIP Model

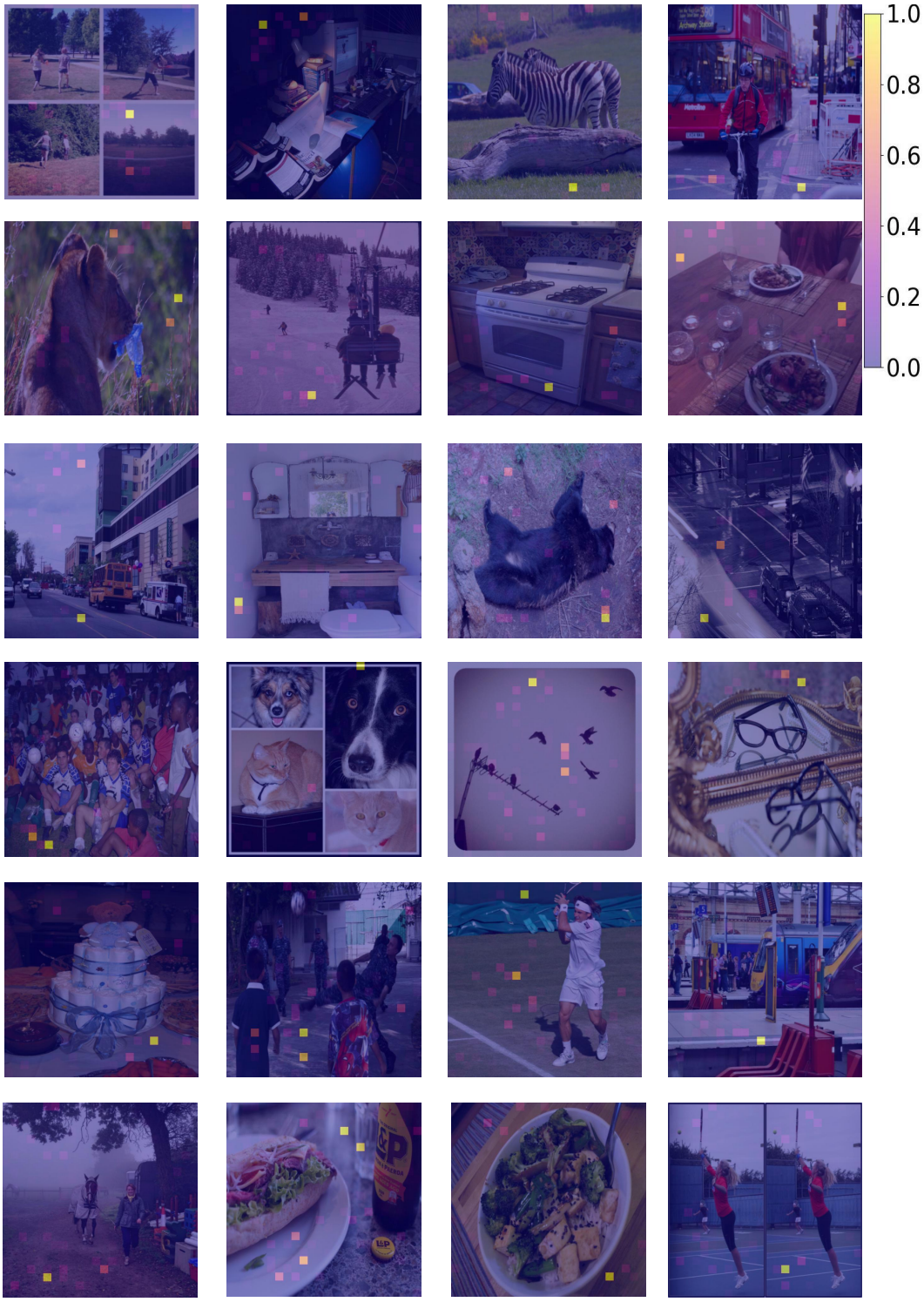


Figure 10. Visualization of Redundancy in the CLIP Model

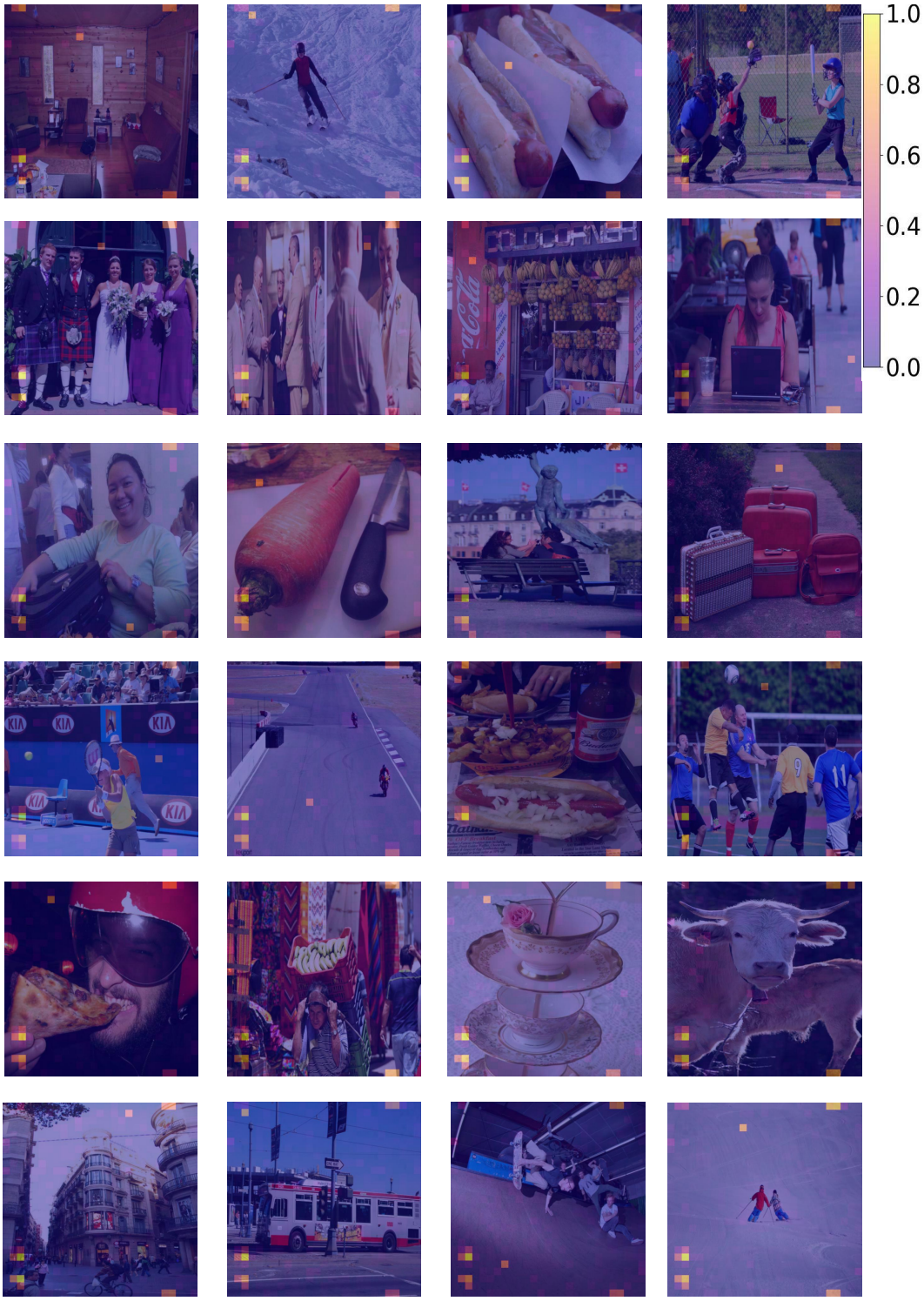


Figure 11. Visualization of Redundancy in the SigLIP Model

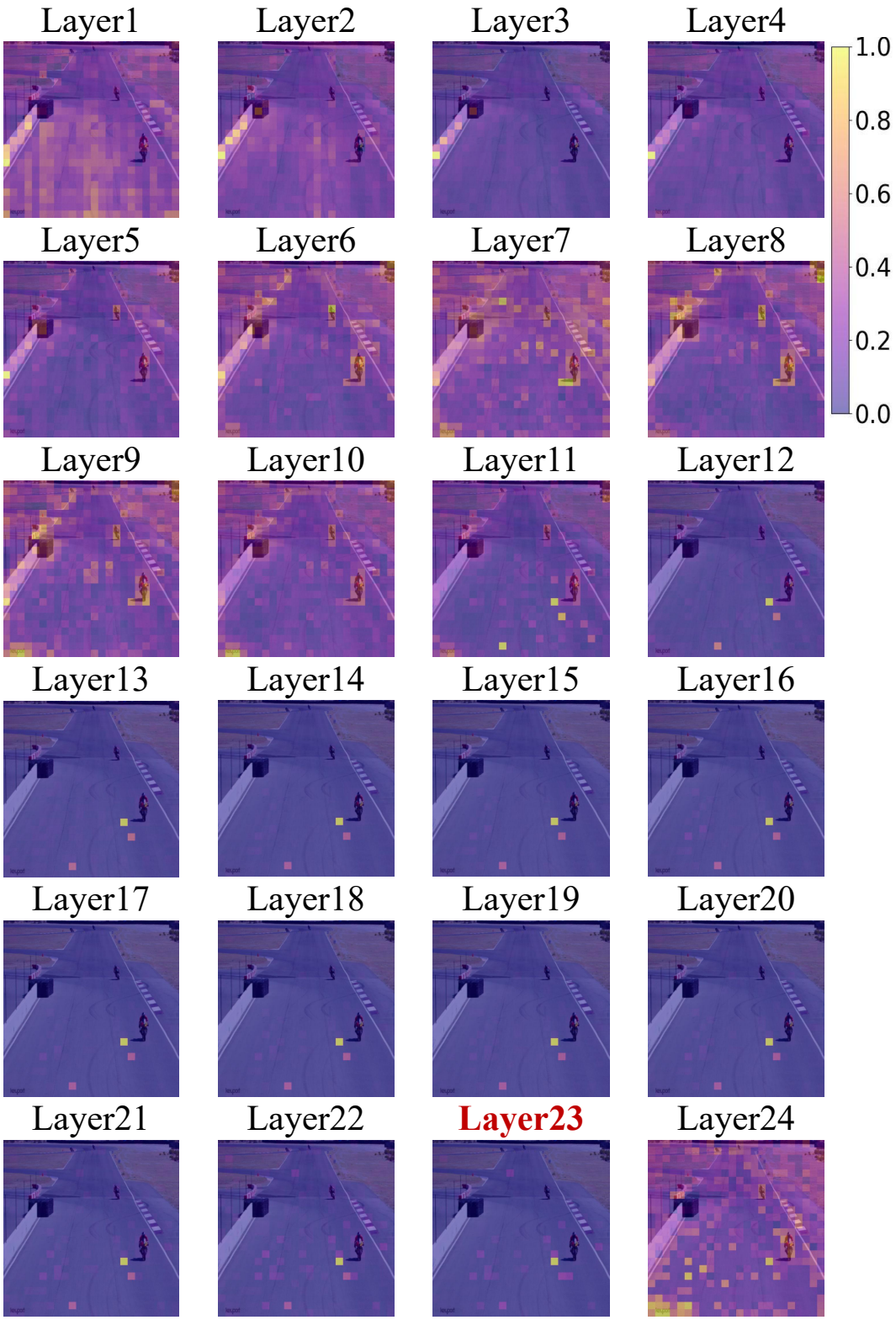


Figure 12. Visualization of Attention Distribution Change

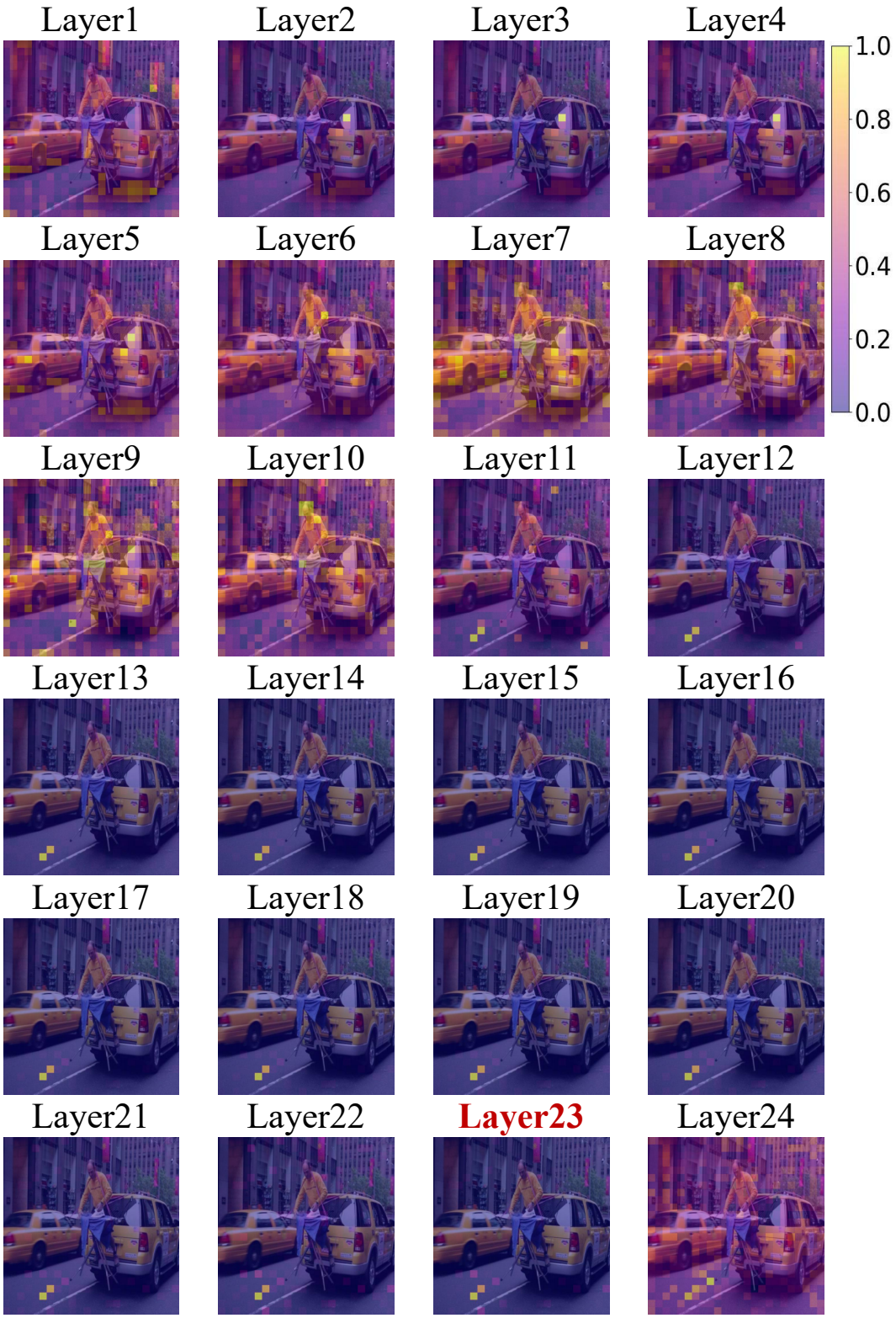


Figure 13. Visualization of Attention Distribution Change

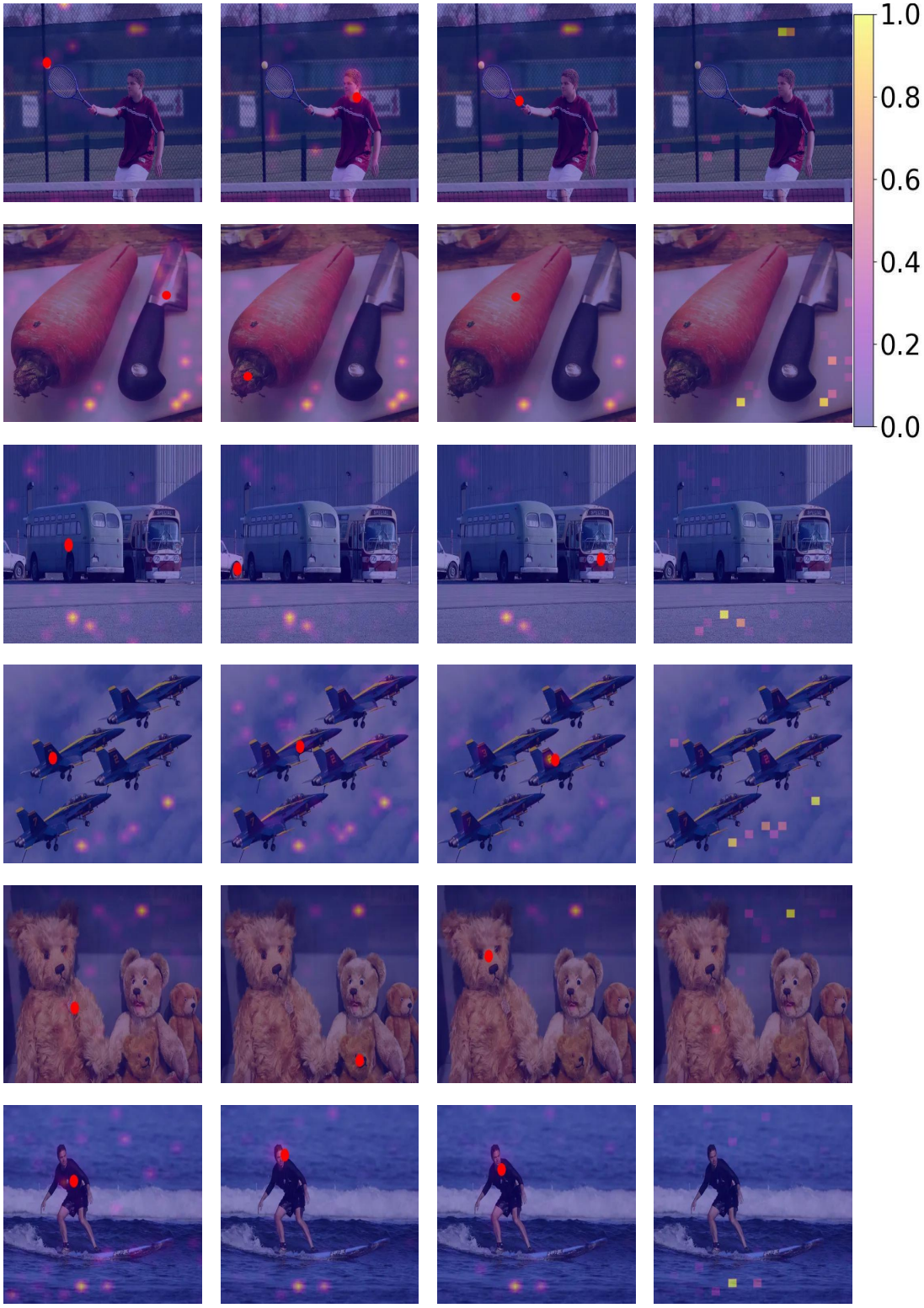


Figure 14. **Visualization of Feature Misalignment.** The red point represents the selected token, while the heatmaps in the first three columns illustrate the attention relationships to the selected token. The last column displays the attention map for the entire image. The results shows that the attention 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

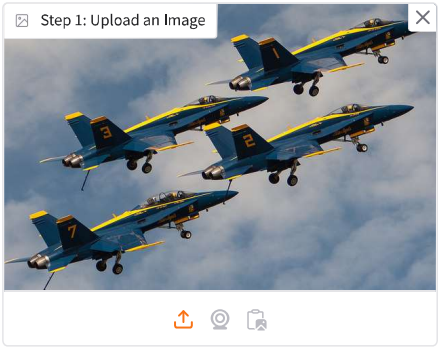
This tool enables the visualization of attention mechanisms in CLIP by analyzing redundancy and feature misalignment in token attention.

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