HIVTP: A Training-Free Method to Improve VLMs Efficiency via Hierarchical Visual Token Pruning Using Middle-Layer-Based Importance Score

Anonymous ACL submission

Abstract

Vision-Language Models (VLMs) have shown strong capabilities on diverse multimodal tasks. However, the large number of visual tokens output by the vision encoder severely hinders inference efficiency, and prior studies have shown that many of these tokens are not important and can therefore be safely pruned. In this work, we propose (**HIVTP**), a training-free method to improve VLMs efficiency via hierarchical visual token pruning using a novel middle-layer-based importance score. Specifically, we utilize attention maps extracted from the middle layers of the vision encoder, which better reflect fine-grained and object-level attention, to estimate visual token importance. Based on this, we propose a hierarchical visual token pruning method to retain both globally and locally important visual tokens. Specifically, we reshape the 1-D visual token sequence output by the vision encoder into a 2-D spatial layout. In the global retaining stage, we divide the image into regions and retain tokens with higher importance scores in each region; in the local retaining stage, we then divide the image into small windows and retain the most important token in each local window. Experimental results show that our proposed method, **HIVTP**, can reduce the time-to-first-token (TTFT) of LLaVA-v1.5-7B and LLaVA-Next-7B by up to 50.0% and 55.1%, respectively, and improve the token generation throughput by up to 60.9% and 47.3%, without sacrificing accuracy, and even achieving improvements on certain benchmarks. Compared with prior works, **HIVTP** achieves better accuracy while offering higher inference efficiency.

1 Introduction

Vision-Language Models (VLMs) such as Gemini Pro (Comanici et al., 2025), GPT-5 (OpenAI, 2025), and LLaVA (Liu et al., 2024a,b) have demonstrated strong capabilities across a wide range of vision-language tasks. They leverage a vision encoder to convert the input image into a sequence of visual tokens, which are then projected into the language feature space. Specifically, these visual tokens are concatenated with textual tokens and jointly processed by a Large Language Model (LLM) (Chiang et al., 2023) to generate responses. Typically, the number of visual tokens is significantly larger than that of textual tokens (Zhang et al., 2025a; Li et al., 2025b,d). For instance, LLaVA-v1.5 (Liu et al., 2024a) converts an input image into 576 visual tokens using CLIP ViT-L/336 (Radford et al., 2021), which is usually more than ten times the number of textual tokens (Li et al., 2025d). This large number of visual tokens hinders the inference efficiency of the LLM (Zhang et al., 2025b,c; Jiang et al., 2024), and prior studies (Jiang et al., 2024; Li et al., 2025a; Yang et al., 2024) have shown that many of these tokens are unimportant for the task at hand and can be removed while maintaining the performance of the VLMs.

Researchers have developed a variety of token compression methods (Li et al., 2023a; Bai et al., 2023; Li et al., 2025c; Cha et al., 2024) that leverage the attention mechanism (Vaswani et al., 2023) or convolution operation (O'Shea and Nash, 2015) to aggregate the visual information from a large number of visual tokens into a few learnable queries. Although these methods reduce the number of visual tokens while preserving the main visual information, they generally require training additional attention layers or convolutional layers, which demands significant additional computational resources.

Several training-free methods have also been proposed (Yang et al., 2024; Jiang et al., 2024; Shang et al., 2024; Arif et al., 2024). Generally, these methods leverage the attention maps from the later layers of the vision encoder to estimate the importance of visual tokens, retain the more important ones, and either discard the less important ones or compress them into a few visual representations.

However, a recent study (Zeng et al., 2024) has shown that in the later layers of the vision encoder, the distribution of attention scores among visual tokens is highly imbalanced: only those encoding abstract high-level semantics receive high attention scores. As a result, existing methods that rely on attention maps from later layers to retain important visual tokens may lose fine-grained visual information, thereby degrading the performance of VLMs on tasks that require detailed visual understanding.

To address the aforementioned limitations, we propose **HIVTP**, a training-free method to improve VLMs efficiency by hierarchically pruning visual tokens using middle-layer-based importance scores. Specifically, we leverage the attention maps from the middle layers of the vision encoder, which have been shown in a recent study (Zeng et al., 2024) to better reflect object-level, fine-grained importance, to compute the importance scores of visual tokens. Then, we propose a hierarchical visual token pruning strategy to retain both globally and locally important visual tokens. We first reshape the 1-D visual token sequence output by the vision encoder into a 2-D spatial layout corresponding to the original image shape. In the global retaining stage, motivated by the insight from (Ye et al., 2024; Wen et al., 2025) that uneven spatial distributions of retained visual tokens after pruning can degrade the performance of VLMs, we evenly divide the image into regions and retain the tokens with higher importance scores from each region. In the local retaining stage, we divide the image into smaller windows and select the token with the highest importance score from the remaining tokens in each window. To evaluate the effectiveness of our method, we apply it to two pioneering VLMs, LLaVA-v1.5-7B (Liu et al., 2024a) and LLaVA-Next-7B (Liu et al., 2024b), and conduct experiments on widely used benchmarks (Fu et al., 2024; Li et al., 2023b; Singh et al., 2019; Liu et al., 2024c; Lu et al., 2022; Liu et al., 2024d; Ainslie et al., 2023; Kembhavi et al., 2016). Our experimental results show that **HIVTP** significantly improves the inference efficiency of both models without sacrificing much accuracy, and even achieves accuracy gains on certain benchmarks. Furthermore, compared with prior methods, **HIVTP** achieves higher accuracy while offering better inference efficiency overall.

Our main contributions are summarized as follows: 1) We propose **HIVTP**, a training-free method that improves the efficiency of VLMs with minimal accuracy loss. 2) We introduce a novel middle-layer-based importance scoring strategy to measure the importance of visual tokens. 3) We propose a hierarchical visual token pruning strategy that retains both globally and locally important visual tokens. 4) Extensive experiments on widely used benchmarks and various VLMs show that **HIVTP** reduces the time-to-first-token (TTFT) by up to 26.5% and increases the token generation throughput by up to 11.1% compared with prior methods, demonstrating the superiority of **HIVTP** over prior methods in improving inference efficiency while preserving accuracy.

2 Related Work

2.1 Vision-Language Models (VLMs)

Closed-source VLMs, such as Gemini Pro (Comanici et al., 2025; Team et al., 2024) and GPT-5 (OpenAI, 2025), are achieving strong performance across a wide range of vision-language tasks. The open-source community has simultaneously made rapid progress. BLIP2 (Li et al., 2023a), MiniGPT-4 (Zhu et al., 2023), and QwenVL (Bai et al., 2023) leverage the Q-Former (Li et al., 2023a) as a projector to compress visual tokens and align vision-text features. In contrast, models such as the LLaVA series (Liu et al., 2024a,b), VILA (Lin et al., 2024), ShareGPT4V (Chen et al., 2024a), CogVLM (Wang et al., 2024), DeepSeekVL (Lu et al., 2024), and InternVL (Chen et al., 2024b) adopt lightweight MLP-based projectors to bridge vision and language representations.

2.2 Visual Token Pruning

Despite the strong performance of current VLMs, their inference efficiency is hindered by the large number of visual tokens, motivating the development of various visual token pruning methods. Q-Former (Li et al., 2023a) and Resampler (Bai et al., 2023) train additional cross-attention layers to aggregate the information from a large number of visual tokens into a few learnable queries. Honeybee (Cha et al., 2024) aggregates local visual tokens into a single token using either convolution mechanisms. However, training additional components leads to substantial computational resource consumption for these methods. FoPru (Jiang et al., 2024) and HiRED (Arif et al., 2024) are training-free methods that utilize attention maps from the penultimate layer of the vision encoder to compute the attention score for each visual token as

its importance score, and prune the less important ones. Building upon similar importance score computing strategies, VisionZip (Yang et al., 2024) and PruMerge (Shang et al., 2024) perform average merging among less important visual tokens and preserve the resulting merged representations. However, these existing training-free methods may lose fine-grained visual information due to the importance scores of visual tokens being computed from the later layers of the vision encoder (Zeng et al., 2024).

3 Preliminaries

VLMs typically comprise three components: a vision encoder, a projector, and a LLM (Chiang et al., 2023). The vision encoder first partitions the input image into $N = n \times n$ non-overlapping patches. Each patch is linearly mapped into a d_v-dimensional embedding and arranged into a sequence. To this sequence, a learnable class token $\text{CLS} \in \mathbb{R}^{1 \times d_v}$ is prepended to facilitate global visual information aggregation, resulting in the visual token matrix $E_v \in \mathbb{R}^{(N+1) \times d_v}$. These $N + 1$ visual tokens are subsequently processed by a set of transformer layers (Vaswani et al., 2023) to facilitate visual information interaction. We denote the N output tokens related to the original image from the penultimate layer of the vision encoder as $Z_v \in \mathbb{R}^{N \times d_v}$. The projector aligns the visual and textual modalities by mapping Z_v into the language space, yielding $V \in \mathbb{R}^{N \times d}$, where d denotes the shared embedding dimensionality between modalities. Meanwhile, the instruction is tokenized and embedded into textual embeddings $T \in \mathbb{R}^{M \times d}$, where M denotes the number of textual tokens. Finally, the visual tokens V and the textual tokens T are concatenated and fed into the LLM, which performs cross-modal interactions through a set of transformer layers and autoregressively generates a textual response R of length L_R,

$$P(R \mid V, T) = \prod_{i=1}^{L_R} P(R_i \mid V, T, R_{<i}) . \quad (1)$$

The prediction of each response token is conditioned on the visual tokens V, the textual tokens T, and previously generated tokens in the response. Consequently, the inference efficiency is heavily affected by the total sequence length, with the number of visual tokens N often far exceeding the number of textual tokens M (Zhang et al., 2025a; Li et al., 2025b,d; Wen et al., 2024; Li et al., 2025a).

4 HIVTP

4.1 Visual Token Importance Score

A common strategy for estimating visual token importance is to measure their average attention scores using attention maps extracted from selected layers of the vision encoder (Jiang et al., 2024).

While previous methods (Yang et al., 2024; Jiang et al., 2024; Shang et al., 2024; Arif et al., 2024) typically select later layers of the vision encoder for this purpose, a recent study (Zeng et al., 2024) has shown that different vison encoder layers focus on different levels of visual semantics. In earlier layers, each visual token tends to receive a similar amount of attention from other visual tokens. In middle layers, visual tokens associated with object-level semantics tend to receive higher attention from other visual tokens, whereas in later layers, tokens encoding abstract high-level visual semantics are more strongly attended. As a result, using attention maps from later layers may degrade performance on tasks that require detailed visual understanding. In Figure 2, we visualize one example from the MME benchmark (Fu et al., 2024), illustrating the attention heatmaps at different layers of the vision encoder in LLaVA-v1.5-7B (Liu et al., 2024a). Brighter regions indicate higher attention scores. This further verifies the distinct attention patterns across different layer depths of the vision encoder.

To avoid performance degradation on fine-grained visual tasks, we capture object-level visual information by extracting attention maps from a set of middle layers in the vision encoder. Specifically, the vision encoder comprises L layers, indexed by $\mathcal{I}(L) = \{1, \ldots, L\}$, with each layer contain H attention heads. We denote the complete collection of multi-head attention maps from all vision encoder layers as,

$$\mathbf{A} = \left\{ \mathbf{A}^{(l,H)} \in \mathbb{R}^{H \times (N+1) \times (N+1)} \,\middle|\, l \in \mathcal{I}(L) \right\},$$

where $\mathbf{A}^{(l,H)}$ represents H attention maps produced in layer l. We define the index set of selected middle layers as $L_s = \{l_m, \cdots, l_{m+q}\} \subseteq \mathcal{I}(L)$, which is identified in Section 5.1. We extract attention maps from each selected layer $l \in L_s$, and define the importance score of each visual token as the average attention it receives from other visual tokens,

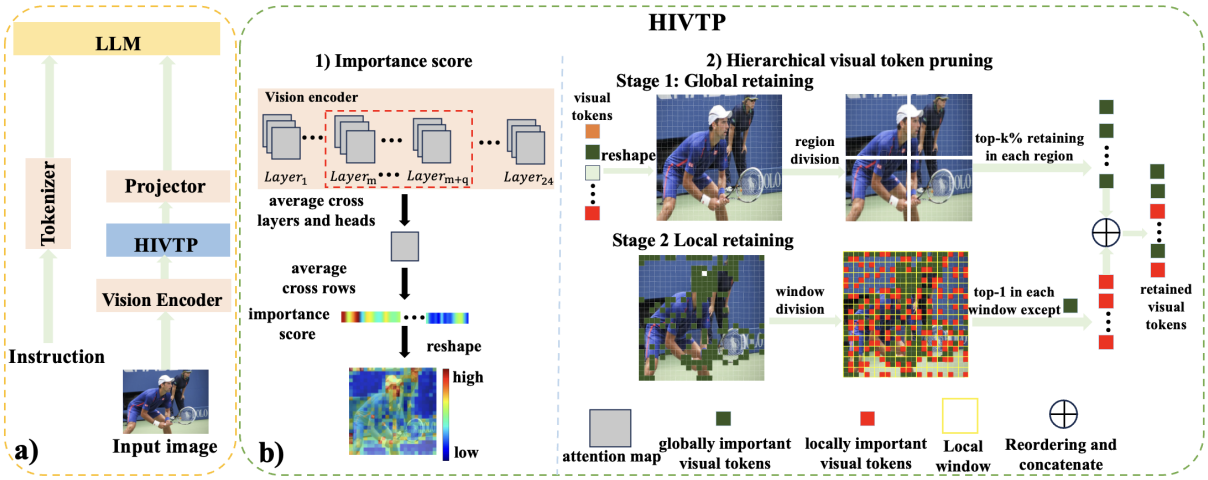

Figure 1: Framework of HIVTP. a) The overall workflow of HIVTP. b) First, we leverage the attention maps from the middle layers of the vision encoder to compute the importance scores of visual tokens. Then, we adopt a hierarchical visual token pruning strategy to retain both globally important and locally important visual tokens.

$$\bar{\mathbf{A}} = \frac{1}{H|L_s|} \sum_{h=1}^{H} \sum_{l \in L_s} \mathbf{A}^{(l,h)} \tag{2}$$

$$\mathbf{S} = \frac{1}{N+1} \sum_{j=0}^{N} \bar{\mathbf{A}}_{j,1:} \tag{3}$$

where $\bar{\mathbf{A}} \in \mathbb{R}^{(N+1)\times(N+1)}$ and $\bar{\mathbf{A}}_{j,1:}$ is the jth row of $\bar{\mathbf{A}}$ excluding the element corresponding to the CLS token. As a result, $\mathbf{S} \in \mathbb{R}^{N\times 1}$ represents the averaged importance scores of the N visual tokens.

4.2 Hierarchical Visual Token Pruning

Our hierarchical visual token pruning strategy first extracts the output visual tokens $\mathbf{Z}_v \in \mathbb{R}^{N\times d_v}$ from the penultimate layer of the vision encoder and compute their corresponding importance scores $\mathbf{S} \in \mathbb{R}^{N\times 1}$ based on the method described in Section 4.1. Subsequently, we perform a global retaining stage and a local retaining stage as described below.

Global Retaining Stage. We aim to select the top-$k\%$ most important globally informative visual tokens from the N visual tokens $\mathbf{Z}_v \in \mathbb{R}^{N\times d_v}$. Previous studies (Ye et al., 2024; Wen et al., 2025) have validated that uneven spatial distributions of preserved visual tokens can degrade the performance of VLMs. Therefore, we uniformly partition the visual tokens into multiple large regions, and perform regional top-$k\%$ selection to enhance the spatial uniformity of the retained tokens.

Specifically, we reshape the visual tokens $\mathbf{Z}_v \in \mathbb{R}^{N\times d_v}$ into a 2-D structure $\mathbf{Z}'_v \in \mathbb{R}^{n\times n\times d_v}$ that reflects the relative spatial positions within the original image. Then, we uniformly divide $\mathbf{Z}'_v$ into r^2 non-overlapping regions and denote the set of regions as $\{\mathcal{R}_i\}_{i=1}^{r^2}$. Each region $\mathcal{R}_i$ contains $N_r = \left(\frac{n}{r}\right)^2$ visual tokens, and we let $\mathcal{I}(\mathcal{R}_i) \subseteq \{1, \ldots, N\}$ denote their positional indices with respect to the original $\mathbf{Z}_v$. For each region $\mathcal{R}_i$, we extract the corresponding importance scores as $\mathbf{S}_i = \mathbf{S}[\mathcal{I}(\mathcal{R}_i)]$. We then select the indices of the top-$k\%$ visual tokens with the highest importance scores in this region based on $\mathbf{S}_i$,

$$\mathcal{I}_i^{\text{global}} = \texttt{TopK}(\mathcal{I}(\mathcal{R}_i), \mathbf{S}_i, k\%). \tag{4}$$

where $\texttt{TopK}(\cdot)$ returns the subset of $\mathcal{I}(\mathcal{R}_i)$ corresponding to the top-$k\%$ indices with the highest importance scores given $\mathbf{S}_i$, and I_i^{global} denotes the index set of globally important visual tokens retained from region $\mathcal{R}_i$. The union of the index sets of globally important visual tokens retained from all regions forms the global visual token set $\mathcal{I}^{\text{global}}$ defined as follows

$$\mathcal{I}^{\text{global}} = \bigcup_{i=1}^{r^2} \mathcal{I}_i^{\text{global}}. \tag{5}$$

We denote the total number of retained globally important visual tokens as $p_g = |\mathcal{I}^{\text{global}}|$, where $|\mathcal{I}^{\text{global}}| = r^2 \cdot N_r \cdot k\%$. Through the global retaining stage, we thus retain globally important visual tokens while ensuring a form of global spatial uniformity.

Local Retaining Stage. To avoid overlooking locally important fine-grained visual tokens, we explicitly perform local spatial modeling to identify these locally important tokens.

Specifically, we partition $\mathbf{Z}'_v \in \mathbb{R}^{n\times n\times d_v}$ into multiple non-overlapping local windows of size

$c \times c$, where c denotes the number of visual tokens along both the height and width of each window, and n is assumed to be divisible by c. As a result, the total number of such windows is denoted by $N_{\mathrm{w}} = \left(\frac{n}{c}\right)^2$, and we denote the set of all windows as $\{\mathcal{W}_j\}_{j=1}^{N_{\mathrm{w}}}$.

We denote the set of positional indices of visual tokens in window $\mathcal{W}_j$ with respect to $\mathbf{Z}_v$ as $\mathcal{I}(\mathcal{W}_j)$. To avoid repeated selections, we exclude tokens that have already been retained in the global stage and define the remaining candidates as $\mathcal{U}_j = \mathcal{I}(\mathcal{W}_j) \setminus \mathcal{I}^{\text{global}}$. If $\mathcal{U}_j$ is non-empty, we collect the index of the visual token with the highest importance score as the locally important visual token in window $\mathcal{W}_j$. Otherwise, no locally important visual token is retained from $\mathcal{W}_j$. This is formalized as follows

$$\mathcal{I}_j^{\text{local}} = \begin{cases} \arg\max_{t \in \mathcal{U}_j} \mathbf{S}[t], & \text{if } \mathcal{U}_j \neq \emptyset, \\ \emptyset, & \text{otherwise}, \end{cases} \quad \forall j \in \{1, \ldots, N_{\mathrm{w}}\}. \tag{6}$$

where I_j^{local} denotes the index set of the locally important visual token retained from the window $\mathcal{W}_j$.

The union of the index sets of locally important visual tokens retained from all windows forms the locally visual token set $\mathcal{I}^{\text{local}}$,

$$\mathcal{I}^{\text{local}} = \bigcup_{j=1}^{N_{\mathrm{w}}} \mathcal{I}_j^{\text{local}}. \tag{7}$$

We denote the length of $\mathcal{I}^{\text{local}}$ as p_l. Since some windows may contribute no tokens (i.e., $\mathcal{U}_j = \emptyset$), it follows that $p_l \leq N_{\mathrm{w}}$.

The index sets of retained globally and locally important visual tokens, $\mathcal{I}^{\text{global}}$ and $\mathcal{I}^{\text{local}}$, are combined to form the final index set $\mathcal{I}^{\text{final}}$ of retained visual tokens,

$$\mathcal{I}^{\text{final}} = \mathcal{I}^{\text{global}} \cup \mathcal{I}^{\text{local}}. \tag{8}$$

To maintain the semantic coherence of retained visual tokens, we sort $\mathcal{I}^{\text{final}}$ in ascending order to obtain the sorted index list $\mathcal{I}^{\text{final}}_{\text{sorted}}$, and extract the corresponding visual tokens from $\mathbf{Z}_v \in \mathbb{R}^{N \times d_v}$ to form the retained list $\mathbf{Z}_v^{\text{retained}} \in \mathbb{R}^{p \times d_v}$, where $p = p_g + p_l$ is the total number of retained tokens. We define the visual token retaining ratio r_{retain} as,

$$r_{\text{retain}} = \frac{p}{N}. \tag{9}$$

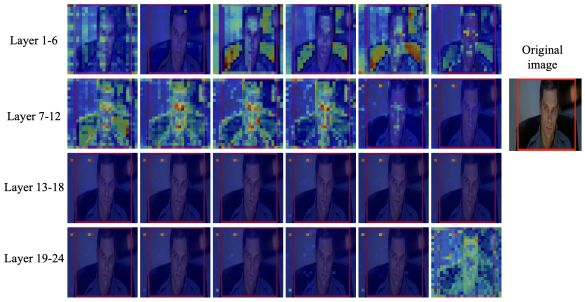

Figure 2: Attention heatmaps at different layers of the vision encoder in LLaVA-v1.5-7B for one example from the MME benchmark. From layer 7 to layer 10, the main object region in the image receives higher attention scores.

Since the number of retained locally important visual tokens satisfies $p_l \leq N_{\mathrm{w}}$, the total number of retained tokens is upper-bounded by $p_{\text{max}} = p_g + N_{\mathrm{w}}$, leading to r_{max} as the upper bound of r_{retain},

$$r_{\text{retain}} \leq r_{\text{max}} = \frac{p_{\text{max}}}{N}. \tag{10}$$

The retained visual tokens $Z_v^{\text{retained}} \in \mathbb{R}^{p \times d_v}$ are projected to align with the language space, and then concatenated with the textual tokens. The overall framework of HIVTP is illustrated in Figure 1 with the number of regions r^2, window size c^2, and r_{max} as hyperparameters.

5 Experimental Results

We evaluate our proposed HIVTP on eight widely used benchmarks when applied to LLaVA-v1.5-7B (Liu et al., 2024a) and LLaVA-Next-7B (Liu et al., 2024b), and compare it with state-of-the-art methods: FoPru (Jiang et al., 2024), VisionZip (Yang et al., 2024), PruMerge (Shang et al., 2024), and HiRED (Arif et al., 2024).

5.1 Experimental Setup

We adopt eight widely used benchmarks, including MME (Fu et al., 2024), POPE (Li et al., 2023b), TextVQA (VQA^{T}) (Singh et al., 2019), MMBench (MMB) (Liu et al., 2024c), ScienceQA (SQA) (Lu et al., 2022), OCRBench (OCR) (Liu et al., 2024d), GQA (Ainslie et al., 2023) and Ai2D (Kembhavi et al., 2016). We conduct experiments under different upper bounds of the visual token keeping ratio, $r_{\text{max}} \in \{75\%, 50\%, 40\%, 25\%\}$. For LLaVA-Next, the original input image is divided into several sub-images. Each sub-image, as well as the original image itself, is independently processed by the vision encoder to obtain its own visual tokens. We apply HIVTP to the visual tokens from both the

original image and all sub-images. For prior methods, we set their actual visual token keeping ratio r_{keep} to match r_{max}. All experiments are conducted using an NVIDIA A40 GPU.

Middle Layers Setting. To determine the index set of middle layers L_s used for computing visual token importance scores, we visualize the attention heatmaps at different layers of the vision encoder in LLaVA-v1.5-7B from the MME benchmark. We observe that in most cases, from layer 7 to 10, the visual tokens corresponding to the main objects receive higher attention. One example is shown in Figure 2. Based on this observation, we select layers 7 to 10 as the middle layers.

5.2 Accuracy Analysis

Accuracy on LLaVA-v1.5. As shown in Table 1, compared to LLaVA-v1.5-7B, under $r_{\text{max}} = 75\%$, 50%, 40%, and 25%, HIVTP achieves average accuracy that is only 0.14%, 0.85%, 1.99%, and 1.63% lower across all benchmarks, respectively. It is worth noting that HIVTP achieves even higher accuracy than LLaVA-v1.5-7B on SQA across all tested values of r_{max}, and also surpasses it on MMB, OCR, and POPE under specific values of r_{max}. Compared to the average score of prior methods, under $r_{\text{max}} = 75\%$, 50%, 40%, and 25%, HIVTP improves accuracy by 2.68%, 2.20%, 1.53%, and 1.42% respectively across all benchmarks, while using even fewer tokens.

Accuracy on LLaVA-Next. As shown in Table 2, compared with LLaVA-Next, HIVTP achieves higher accuracy on both the MME and SQA benchmarks under all values of r_{max}, with improvements of up to 1.7% and 1.0%, respectively. On the remaining benchmarks, HIVTP exhibits only a slight decrease of 0.8% on average across all values of r_{max}. Besides, compared to the average score of prior methods, under $r_{\text{max}} = 75\%$, 50%, 40%, and 25%, HIVTP improves accuracy by 3.10%, 2.92%, 2.73%, and 3.43% respectively across all benchmarks, while using fewer tokens.

Discussion. Overall, on both LLaVA-v1.5-7B and LLaVA-Next, HIVTP achieves higher accuracy while using fewer visual tokens compared to prior methods. Compared to the original LLaVA-v1.5-7B and LLaVA-Next models, implementing HIVTP can prune a large number of visual tokens without sacrificing much accuracy. It consistently achieves higher accuracy on SQA. We hypothesize that this is because the task is less dependent on visual information, and, by pruning less important visual tokens, performance can be improved to a certain extent.

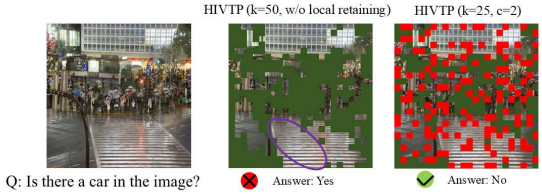

Figure 3: HIVTP ($k = 50$, w/o local retaining) hallucinates a car in the image. In contrast, HIVTP ($k = 25$, c=2) avoids hallucination. Green areas indicate the retained globally important visual tokens, while red areas indicate the retained locally important visual tokens. The region marked by the purple ellipse highlights a large contiguous area of pruned tokens.

5.3 Efficiency Analysis

To quantify the advantage of HIVTP in terms of inference efficiency, we adopt two widely used metrics: time-to-first-token (TTFT) (Jiang et al., 2024)and token generation throughput (throughput) (Arif et al., 2024).We compute the average results of HIVTP and prior methods across all benchmarks, and the results on LLaVA-v1.5-7B and LLaVA-Next are presented in Table 1 and Table 2, respectively.

HIVTP significantly improves the inference efficiency of both LLaVA-v1.5-7B and LLaVA-Next-7B. Compared to the original models, at the minimum tested r_{max} of 25%, HIVTP reduces TTFT by up to 50.0% and 55.1%, respectively, and increases throughput by up to 60.9% and 47.3%, respectively. Additionally, HIVTP achieves higher inference efficiency than prior methods overall, mainly because it retains fewer visual tokens under the same r_{max}. Compared to VisionZip, HIVTP reduces TTFT by 26.5% under $r_{\text{max}} = 40\%$ on LLaVA-v1.5-7B. Compared to PruMerge, HIVTP increases throughput by 11.1% under $r_{\text{max}} = 40\%$ on LLaVA-Next-7B. On LLaVA-v1.5-7B, the average TTFT of HIVTP across different r_{max} values is 3.79% lower than that of FoPru, which has the lowest average TTFT among prior methods. The average throughput of HIVTP exceeds that of HiRED, the highest among prior methods, by 3.12%. On LLaVA-Next-7B, HIVTP achieves an average TTFT that is 3.41% lower and an average throughput that is 1.27% higher than HiRED, which has the lowest TTFT and highest throughput among prior methods.

Model & Method	r_{retain}	# Visual tokens	Accuracy Performance								Efficiency	
			MME ↑	POPE ↑	VQA^{T} ↑	MMB ↑	SQA ↑	OCR ↑	GQA ↑	Ai2D ↑	TTFT (ms) ↓	Throughput (token/second) ↑
r_{max} = 100% (576 Visual Tokens)												
LLaVA-v1.5-7B	100.0%	576	1513.51	86.98	46	64.7	69.51	31.2	61.97	55.5	140	18.66
r_{max} = 75% (432 Visual Tokens)												
FoPru	75.0%	432	1508.98	86.51	45.40	64.18	70.02	31.00	53.41	54.07	121	21.95
VisionZip	75.0%	432	1497.65	86.82	41.08	64.00	70.10	29.80	59.48	54.92	119	22.49
PruMerge	75.0%	432	1462.31	84.78	39.43	61.86	**70.12**	28.70	60.57	53.92	124	21.8
HiRED	75.0%	432	1502.73	86.93	45.27	63.40	69.58	30.90	60.60	54.40	117	22.75
HIVTP (Ours) (r=2, k=50, c=2)	70.1%	404	**1512.41**	**87.20**	**45.44**	**65.03**	69.89	**31.40**	**61.42**	**54.98**	**114**	**22.95**
r_{max} = 50% (288 Visual Tokens)												
FoPru	50.0%	288	1445.73	84.89	45.55	62.80	69.84	30.00	59.80	54.07	100	22.59
VisionZip	50.0%	288	1464.15	86.51	41.07	**64.09**	69.89	29.50	59.18	54.92	97	23.14
PruMerge	50.0%	288	1432.62	85.13	39.53	61.77	69.89	29.40	60.49	53.72	101	22.39
HiRED	50.0%	288	1474.46	87.04	45.46	62.97	69.70	30.00	60.50	54.88	100	22.04
HIVTP (Ours) (r=2, k=25, c=2)	48.9%	282	**1495.37**	**87.12**	**45.84**	64.00	**70.12**	**30.50**	**60.66**	**54.99**	**92**	**23.7**
r_{max} = 40% (230 Visual Tokens)												
FoPru	40.0%	230	1445.51	87.01	44.91	61.32	69.37	29.40	59.87	54.47	75	26.71
VisionZip	40.0%	230	1451.11	86.79	41.43	62.12	69.72	29.60	58.94	54.76	102	25.20
PruMerge	40.0%	230	1444.13	85.23	39.64	61.77	69.63	28.40	59.32	54.92	90	26.80
HiRED	40.0%	230	1442.41	86.83	**45.22**	61.49	69.75	30.10	59.94	54.92	80	27.94
HIVTP (Ours) (r=2, k=15, c=2)	39.2%	226	**1454.26**	**87.07**	45.04	**62.20**	**70.01**	**30.20**	**60.20**	**55.02**	**75**	**27.99**
r_{max} = 25% (144 Visual Tokens)												
FoPru	25.0%	144	1400.14	85.01	44.25	61.86	69.72	**30.50**	57.44	54.73	71	29.91
VisionZip	25.0%	144	1417.35	85.04	41.09	60.74	69.70	28.00	57.05	54.16	72	29.71
PruMerge	25.0%	144	1400.91	84.37	38.80	60.22	69.77	28.00	58.06	54.37	74	29.46
HiRED	25.0%	144	**1436.28**	85.13	44.24	61.43	69.65	28.40	58.31	54.89	77	28.89
HIVTP (Ours) (r=2, k=14, c=3)	24.6%	142	1403.48	**86.80**	**44.40**	**61.99**	**69.94**	28.80	**58.70**	**54.92**	**70**	**30.02**

Table 1: Accuracy and efficiency comparison of HIVTP with prior methods on LLaVA-v1.5-7B under $r_{\text{max}} \in \{75\%, 50\%, 40\%, 25\%\}$. The actual retaining ratio is denoted by r_{retain}. TTFT and Throughput are computed as averages across all benchmarks.

6 Ablation

We evaluate the influence of key components of HIVTP on model performance. For fairness, all experiments are conducted on the LLaVA-v1.5-7B using the MME, POPE, and Ai2D benchmarks, with $r_{\text{max}} = 50\%$.

Influence of Layers. As shown in Table 3, using the middle layers of the vision encoder, $L_s = \{l_7, l_8, l_9, l_{10}\}$, to compute the importance scores yields better performance compared to using later layers $L_s = \{l_{23}\}$ (Yang et al., 2024; Jiang et al., 2024; Shang et al., 2024; Arif et al., 2024), $L_s = \{l_{20}, l_{21}, l_{22}, l_{23}\}$, and earlier layers $L_s = \{l_1, l_2, l_3, l_4\}$. This result may be attributed to the fact that attention maps from middle layers are more effective at identifying object-level visual tokens, whereas attention maps from later layers and earlier layers can only identify high-level semantic tokens and struggle to estimate visual token importance, respectively.

Influence of Global Region Size. We further evaluate the influence of region division in the global retaining stage on model performance. HIVTP (r=1, w/o region division) refers to retaining the top-25% important tokens directly from all visual tokens without region division, and applying 2×2 windows in the local retaining stage. It performs worse than those that incorporate region division. This may be because the retaining process in this case relies solely on visual importance scores without explicitly considering the spatial distribution of tokens, resulting in globally important visual tokens being unevenly distributed and thus degrading performance. However, dividing the image into 9 global regions (HIVTP (r=3)) performs worse than dividing it into 4 global regions (HIVTP (r=2)), possibly because excessive region division leads to too few visual tokens within each region, making it difficult to retain globally representative tokens with high importance scores.

Influence of Window Division. HIVTP ($k = 50$, w/o local retaining) refers to the hierarchical visual token pruning strategy that retains the top-50% important tokens from each of the four regions in the global retaining stage only. HIVTP ($k = 39$, c=3) and HIVTP ($k = 25$, c=2) refer to retaining the top-39% and top-25% important tokens from each of the four regions and applying 3×3 and 2×2 windows in the local retaining stage, respectively. We set their k to 39 and 25 to ensure that $r_{\text{max}} = 50\%$. As shown in Table 3, HIVTP with local retaining consistently outperforms the variant without local retaining, demonstrating the benefit of retaining locally important fine-grained visual tokens. In particular, HIVTP with local retaining achieves better performance on the POPE benchmark, further indicating that local retaining helps

Model & Method	r_{retain}	# Visual tokens	Accuracy Performance								Efficiency	
			MME ↑	POPE ↑	VQA^{T} ↑	MMB ↑	SQA ↑	OCR ↑	GQA ↑	Ai2D ↑	TTFT (ms) ↓	Throughput (token/second) ↑
r_{max} = 100% (2614 Visual Tokens)												
LLaVA-Next-7B	100.0%	2614	1484.20	87.74	64.78	67.87	78.19	50.2	64.35	66.32	423	13.28
r_{max} = 75% (1959 Visual Tokens)												
FoPru	75.0%	1959	1492.66	87.99	64.73	67.35	78.00	50.60	64.09	66.06	353	14.77
VisionZip	75.0%	1959	1479.59	87.99	64.45	67.53	78.00	50.00	64.10	65.93	343	14.88
PruMerge	75.0%	1959	1270.49	79.53	50.79	62.80	78.68	39.20	61.72	63.41	344	15.09
HiRED	75.0%	1959	1478.38	88.01	64.50	67.53	78.09	**50.80**	64.27	66.13	345	14.96
HIVTP (Ours) (r=2, k=50, c=2)	71.7%	1857	**1509.23**	**88.46**	**64.75**	**67.99**	**78.95**	49.30	**64.31**	**66.89**	**324**	**15.47**
r_{max} = 50% (1307 Visual Tokens)												
FoPru	50.0%	1307	1485.38	87.23	62.13	65.53	78.31	47.90	63.25	65.45	243	17.78
VisionZip	50.0%	1307	1486.58	88.28	63.53	67.53	77.98	50.50	63.75	65.84	244	17.50
PruMerge	50.0%	1307	1286.08	79.33	50.03	62.63	78.57	38.40	61.73	63.86	247	17.65
HiRED	50.0%	1307	1474.85	88.23	63.15	67.78	78.12	**51.00**	63.83	65.74	246	17.68
HIVTP (Ours) (r=2, k=25, c=2)	49.3%	1291	**1487.49**	**88.34**	**63.80**	**67.97**	**78.92**	48.40	**64.03**	**66.06**	**242**	**17.85**
r_{max} = 40% (1043 Visual Tokens)												
FoPru	40%	1043	1481.96	87.94	62.90	66.67	78.05	**50.90**	63.45	65.05	220	18.49
VisionZip	40.0%	1043	1484.58	87.79	63.23	66.58	78.17	49.80	63.58	65.65	237	18.15
PruMerge	40.0%	1043	1270.50	79.12	49.32	61.34	78.16	37.70	61.32	63.86	248	16.97
HiRED	40.0%	1043	1469.12	88.00	63.53	67.44	77.95	50.70	63.66	65.68	216	18.79
HIVTP (Ours) (r=2, k=15, c=2)	39.4%	1031	**1490.98**	**88.03**	**63.61**	**67.46**	**78.24**	48.50	**63.87**	**65.77**	**215**	**18.86**
r_{max} = 25% (653 Visual Tokens)												
FoPru	25.0%	653	1474.91	86.56	62.23	65.81	78.02	48.10	61.87	65.48	200	18.86
VisionZip	25.0%	653	1459.96	87.31	62.10	65.46	78.47	48.20	62.89	65.61	233	18.84
PruMerge	25.0%	653	1234.16	79.08	48.50	59.54	78.61	37.60	60.38	63.67	191	**20.68**
HiRED	25.0%	653	1467.24	87.36	62.28	65.98	78.38	**48.60**	63.24	65.67	201	19.47
HIVTP (Ours) (r=2, k=14, c=3)	24.8%	650	**1493.89**	**87.69**	**62.79**	**66.52**	**78.73**	47.44	**63.65**	**65.87**	**190**	19.54

Table 2: Accuracy and efficiency comparison of HIVTP with prior methods on LLaVA-Next-7B under $r_{\text{max}} \in \{75\%, 50\%, 40\%, 25\%\}$. The actual retaining ratio is denoted by r_{retain}. TTFT and Throughput are computed as averages across all benchmarks.

Choice	MME ↑	POPE ↑	Ai2D ↑
Influence of Layers ($r = 2, k = 25, c = 2, r_{max} = 50\%$)			
HIVTP($L_s = \{l_{23}\}$)	1480.08	86.15	54.06
HIVTP($L_s = \{l_{20}, l_{21}, l_{22}, l_{23}\}$)	1481.49	87.02	54.76
HIVTP($L_s = \{l_7, l_8, l_9, l_{10}\}$)	**1495.37**	**87.12**	**54.99**
HIVTP($L_s = \{l_1, l_2, l_3, l_4\}$)	1410.26	84.60	54.21
Influence of Region Division ($k = 25, c = 2, r_{max} = 50\%, L_s = \{l_7, l_8, l_9, l_{10}\}$)			
HIVTP(r=1, w/o region division)	1469.70	86.39	54.63
HIVTP(r=2)	**1495.37**	**87.12**	**54.99**
HIVTP(r=3)	1476.76	86.78	54.89
Influence of Window Division ($r = 2, r_{max} = 50\%, L_s = \{l_7, l_8, l_9, l_{10}\}$)			
HIVTP($k = 50$, w/o local retaining)	1453.66	86.92	54.84
HIVTP($k = 25$, c=2)	**1495.37**	**87.12**	**54.99**
HIVTP($k = 39$, c=3)	1473.86	87.04	54.93

Table 3: Ablation study on key components in HIVTP.

mitigate hallucination. We hypothesize that this improvement stems from the reduced visual uncertainty, as prior studies (Ding et al., 2024; Leng et al., 2023) have shown that visual uncertainty amplifies hallucination in VLMs. To empirically validate our hypothesis, we visualize an example from the POPE benchmark to illustrate the difference between HIVTP ($k = 50$, w/o local retaining) and HIVTP ($k = 25$, c=2) in Figure 3. Compared to HIVTP ($k = 50$, w/o local retaining), HIVTP ($k = 25$, c=2) promotes a more spatially uniform retention of visual tokens, effectively avoiding the dropping of large contiguous regions and thereby reducing visual uncertainty. This helps avoid hallucinating a car in the image.

7 Conclusion and Limitations

This paper proposes HIVTP, a training-free method to improve VLMs inference efficiency via hierarchical visual token pruning using a novel middle-layer-based importance score. We leverage attention maps from selected middle layers of the vision encoder to estimate the importance of visual tokens, as these layers better capture fine-grained and object-level attention. We then propose a hierarchical visual token pruning strategy. In the global retaining stage, we perform region division and retain tokens with higher importance scores in each region to obtain globally important visual tokens. In the local retaining stage, we perform window division and retain the token with the highest importance score in each window to obtain locally important visual tokens. Extensive evaluations on two LLaVA models using widely adopted benchmarks demonstrate that HIVTP can significantly improve inference efficiency with minimal accuracy loss, and even improve accuracy on certain benchmarks.

Despite the strong performance demonstrated by HIVTP, it still has limitations. Specifically, the token retaining ratio in our method is manually set, whereas the optimal ratio may vary across different visual tasks. Future work could explore how to dynamically adjust the token retaining ratio in a task-wise manner.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. *Preprint*, arXiv:2305.13245.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. 2024. HiRED: Attention-guided token dropping for efficient inference of high-resolution vision-language models. *Preprint*, arXiv:2408.10945.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *Preprint*, arXiv:2308.12966.

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

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2024a. ShareGPT4V: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024b. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Peng Ding, Jingyu Wu, Jun Kuang, Dan Ma, Xuezhi Cao, Xunliang Cai, Shi Chen, Jiajun Chen, and Shujian Huang. 2024. Hallu-PI: Evaluating hallucination in multi-modal large language models within perturbed inputs. *Preprint*, arXiv:2408.01355.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2024. MME: A comprehensive evaluation benchmark for multimodal large language models. *Preprint*, arXiv:2306.13394.

Lei Jiang, Weizhe Huang, Tongxuan Liu, Yuting Zeng, Jing Li, Lechao Cheng, and Xiaohua Xu. 2024. FoPru: Focal pruning for efficient large vision-language models. *Preprint*, arXiv:2411.14164.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. *Preprint*, arXiv:1603.07396.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2023. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. *Preprint*, arXiv:2311.16922.

Duo Li, Zuhao Yang, and Shijian Lu. 2025a. ToDRE: Visual token pruning via diversity and task awareness for efficient large vision-language models. *Preprint*, arXiv:2505.18757.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Kevin Y. Li, Sachin Goyal, Joao D. Semedo, and J. Zico Kolter. 2025b. Inference optimal VLMs need fewer visual tokens and more parameters. *Preprint*, arXiv:2411.03312.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. 2025c. TokenPacker: Efficient visual projector for multimodal LLM. *International Journal of Computer Vision*, pages 1–19.

Yifan Li, Wentao Bao, Botao Ye, Zhen Tan, Tianlong Chen, Huan Liu, and Yu Kong. 2025d. Window token concatenation for efficient visual large language models. *Preprint*, arXiv:2504.04024.

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

Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. 2024. VILA: On pre-training for visual language models. *Preprint*, arXiv:2312.07533.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. *Preprint*, arXiv:2310.03744.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024b. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2024c. MMBench: Is your multi-modal model an all-around player? *Preprint*, arXiv:2307.06281.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. 2024d. OCRBench: on the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12).

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, and 1 others. 2024. DeepSeek-VL: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Preprint*, arXiv:2209.09513.

OpenAI. 2025. Introducing GPT-5. `https://openai.com/index/introducing-gpt-5/`. Published August 7, 2025. Accessed: 2025-09-17.

Keiron O'Shea and Ryan Nash. 2015. An introduction to convolutional neural networks. *Preprint*, arXiv:1511.08458.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2024. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. *Preprint*, arXiv:2403.15388.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards VQA models that can read. *Preprint*, arXiv:1904.08920.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, and 1 others. 2024. CogVLM: Visual expert for pretrained language models. *Advances in Neural Information Processing Systems*, 37:121475–121499.

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

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. 2025. Token pruning in multimodal large language models: Are we solving the right problem? *Preprint*, arXiv:2502.11501.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2024. VisionZip: Longer is better but not necessary in vision language models. *Preprint*, arXiv:2412.04467.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. 2024. ATP-LLaVA: Adaptive token pruning for large vision language models. *Preprint*, arXiv:2412.00447.

Jingjie Zeng, Zhihao Yang, Qi Yang, Liang Yang, and Hongfei Lin. 2024. Peeling back the layers: Interpreting the storytelling of ViT. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7298–7306.

Hao Zhang, Mengsi Lyu, Chenrui He, Yulong Ao, and Yonghua Lin. 2025a. Towards adaptive visual token pruning for large multimodal models. *Preprint*, arXiv:2509.00320.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. 2025b. Beyond text-visual attention: Exploiting visual cues for effective token pruning in VLMs. *Preprint*, arXiv:2412.01818.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. 2025c. SparseVLM: Visual token sparsification for efficient vision-language model inference. *Preprint*, arXiv:2410.04417.

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