This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

One Layer's Trash is Another Layer's Treasure: Adaptive Layer-wise Visual Token Selection in LVLMs

Yongru Chen[1*] Kai Zhang[3,1*] Zeliang Zong[1] Yuchen Lu[2]
Wenming Tan[1†] Ye Ren[1] Jilin Hu[3†]
[1]Hikvision Research Institute [2]Peking University [3]East China Normal University

Abstract

*Large Vision-Language Models (LVLMs) have achieved remarkable success across diverse multimodal tasks, yet their practical deployment remains constrained by the computational burden arising from lengthy visual tokens. While visual token pruning has emerged as a promising solution, existing methods suffer from a fundamental limitation: once tokens are pruned at a specific layer, they become inaccessible to all subsequent layers, leading to premature information loss that can compromise model performance. Through empirical studies, we observe that different layers exhibit distinct visual region focus, indicating a varying optimal token subset across layers. Motivated by this insight, we propose **A**daptive **L**ayer-wise **V**isual **T**oken **S**election (ALVTS), a novel framework that breaks away from the conventional static token pruning paradigm. ALVTS incorporates a lightweight token selector to identify and route important tokens for further processing, while allowing less important tokens to skip the layer, thus minimizing computational redundancy. These two streams of tokens are seamlessly reintegrated before being fed into subsequent layers, facilitating adaptive compression across the entire model. Grounded in our importance consistency constrained low-rank approximation, the proposed token selection module closely emulates the full attention mechanism, effectively capturing its essential patterns without requiring model retraining. Extensive experiments on LLaVA-1.5, LLaVA-NeXT, and Qwen2.5-VL validate the effectiveness of our method. With an 89% token compression ratio, ALVTS retains 96.7% of the original model's accuracy, achieving a superior efficiency-accuracy trade-off for LVLM inference.*

1. Introduction

The remarkable success of Large Language Models (LLMs) [6, 35, 42, 50] in natural language processing has catalyzed the development of Large Vision-Language Models (LVLMs) [5, 19, 24, 38], which aim to bridge visual and textual modalities for a wide range of multimodal tasks, such as visual question answering [4, 14, 26], image captioning [7, 8, 27], and object grounding [25, 28, 41]. While LVLMs have demonstrated impressive performance, the substantial number of visual tokens poses significant computational challenges. For instance, LLaVA-NeXT [24] can generate up to 2880 tokens when processing a 672×672 image, whereas textual inputs typically contain fewer than 100 tokens. This imbalance results in a disproportionate computational load dominated by the visual modality [32, 47], which significantly limits the practical deployment of LVLMs, especially in resource-constrained environments [33, 44].

Previous research [9, 21] has demonstrated that visual tokens in VLMs exhibit significant redundancy, and visual token pruning has emerged as a promising solution to address the computational bottleneck of VLMs. Many approaches achieve this by estimating token importance from attention signals at specific modules. Common choices include the [CLS] token attention [37, 43, 49] in the ViT backbone [12], or the text-to-vision attention [9, 45, 46] in the LLM component. While intuitive, these methods share a common limitation: once tokens are pruned at a specific layer, they become inaccessible to all subsequent layers. This leads to the irrecoverable loss of visual information that may prove crucial in later layers, thereby degrading the model's generation quality.

We illustrate this issue using the pioneering FastV [9] algorithm as an example. FastV prunes tokens based on attention scores derived from the layer 2 of the LLM. Following this methodology, we visualize the image tokens that receive high attention at each layer, as shown in Fig. 1. We observe clear cross-layer heterogeneity: layer 2 attends predominantly to the lower bridge structure, layer 10 captures the bridge name plaque, and layer 20 concentrates on the tower structure, demonstrating distinct attention patterns across layers. Only by integrating important visual information across multiple layers can the model generate accurate

*Equal contribution. † Corresponding authors.

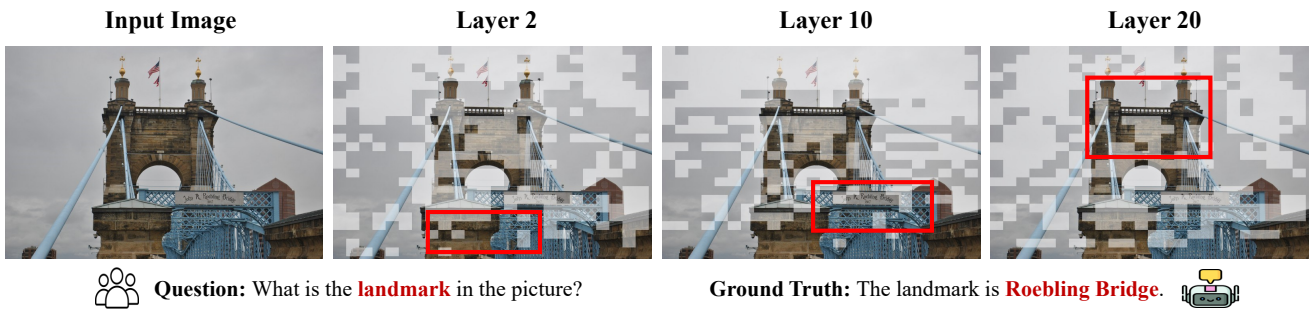


Figure 1. Layer-wise visual token attention patterns of FastV. Colored and grey regions indicate important and unimportant tokens, respectively. Red boxes highlight representative attention regions in each layer.

responses. Conversely, pruning tokens at a single layer risks discarding instruction-relevant regions, leading to incorrect predictions.

Motivated by the above observation, we propose ALVTS, a novel and efficient framework for visual token compression. Unlike existing approaches that retain a fixed subset of compressed tokens across all layers, ALVTS performs layer-wise dynamic token selection. By allowing each layer to process only the most relevant tokens for its specific visual focus, we can dramatically reduce redundant computations without sacrificing the model's ability to capture essential visual semantics.

A key challenge in layer-wise token selection is determining which tokens are most important without performing full attention computations, as this would incur excessive computational overhead [2, 16]. To address this, we design a lightweight token selector that performs dynamic token selection at each layer (see Fig. 2). Specifically, before visual tokens are fed into a decoder layer, they are first processed by the selector which computes importance scores for each token. Then we dynamically select a subset of high-scoring tokens to participate in the current layer's computation, while low-importance tokens skip the layer directly, avoiding unnecessary operations. At the output of the decoder layer, the processed tokens are seamlessly merged with the skipped tokens, restoring the complete token sequence for subsequent layers. This process repeats independently at each layer, enabling adaptive and context-aware token pruning throughout the model. Crucially, our token selector is theoretically grounded in importance consistency constrained low-rank approximation, which enables it to closely emulate the full attention mechanism in token selection behavior. This design eliminates the need for retraining the models, making ALVTS practical for real-world deployment.

To validate the effectiveness of ALVTS, we conduct extensive experiments on popular LVLM architectures, including LLaVA-1.5 [23], LLaVA-NeXT [24], and Qwen2.5-VL [5]. Our experimental results demonstrate that ALVTS achieves a favorable balance between efficiency and accuracy. When compressing 89% of visual tokens, ALVTS maintains 96.7% of the original model's performance and delivers a 1.6× inference speedup.

Our main contributions are summarized as follows:

- We reveal a fundamental limitation in existing token pruning strategies: once tokens are pruned at a specific layer, they become inaccessible to all subsequent layers, leading to irreversible information loss that can be critical for deeper layers.
- We introduce ALVTS, which performs adaptive visual token selection at each layer rather than using a globally fixed token subset. This approach eliminates redundant computations while preserving essential visual semantics throughout the model.
- We develop an efficient low-rank token selector that maintains high fidelity to the full attention mechanism without requiring model retraining.
- Comprehensive evaluations across various LVLMs and benchmarks demonstrate that ALVTS achieves superior efficiency-accuracy trade-offs.

2. Related Works

2.1. Large Vision-Language Models

Several representative architectures have advanced the field of Large Vision-Language Models significantly. LLaVA [23] combines CLIP vision encoders [30] with language models through trainable projection layers, demonstrating effective vision-language alignment via instruction tuning. LLaVA-NeXT [24] introduces the AnyRes technique to process high-resolution images, balancing visual detail preservation with computational efficiency. Qwen2.5-VL [5] employs the naive dynamic resolution mechanism that can handle arbitrary image resolutions, which ensures consistency between the model input and the inherent information in images. The InternVL series [10, 38] divides images into a variable number of tiles

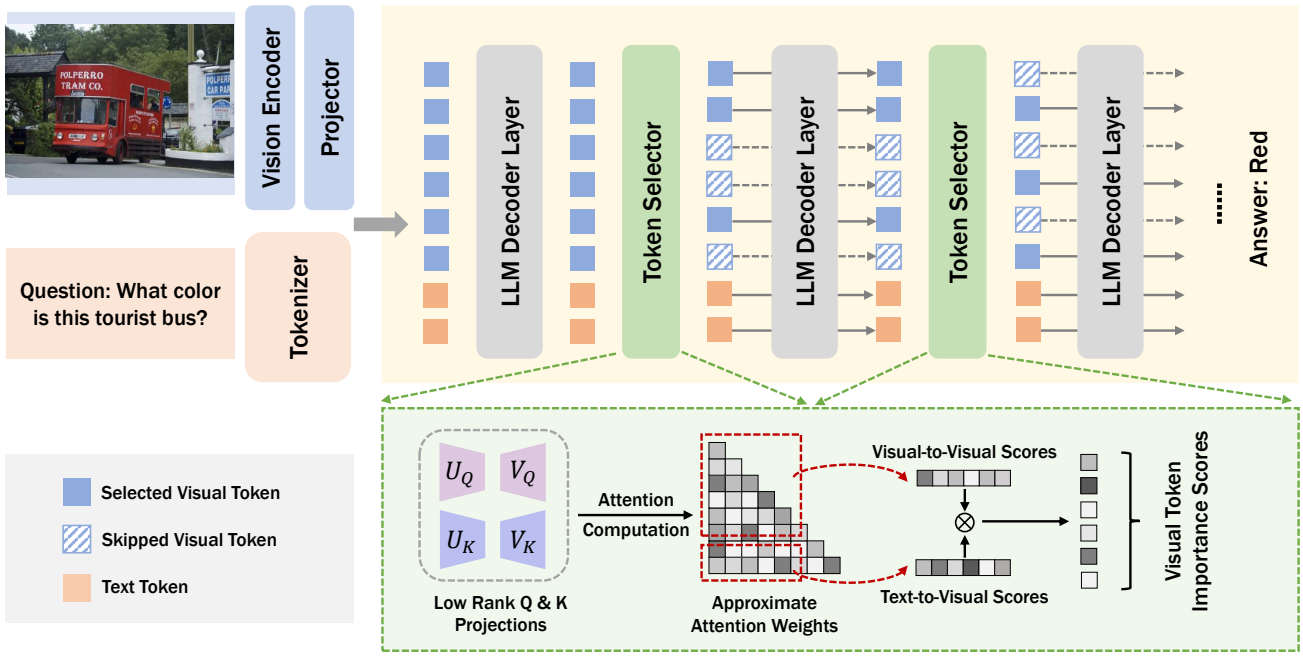


Figure 2. Overview of ALVTS. The framework performs adaptive layer-wise visual token selection. At each LLM decoder layer, a lightweight token selector computes importance scores using low-rank approximations of full attention mechanism. Important tokens participate in the layer's computation while less important tokens skip the layer. Both streams of tokens are reintegrated before the next layer, enabling adaptive compression across the model.

according to aspect ratio, supporting up to 4K resolution input. Despite these impressive capabilities, LVLMs face substantial efficiency challenges as the number of visual tokens increases dramatically [32].

2.2. Visual Token Compression for LVLMs

Existing methods primarily fall into attention-based and diversity-based categories. Attention-based approaches leverage attention mechanisms [36] to assess token importance and guide pruning decisions. Methods utilizing vision encoder attention, such as VTC-CLS [37], VisPruner [49], and VisionZip [43], exploit [CLS] token attention to identify dominant tokens that aggregate substantial visual information. Another category of methods exploits attention within language models to evaluate token importance. For instance, FastV [9] prunes visual tokens with low attention scores after the filtering layer. PyramidDrop [40] observes that attention to visual tokens progressively concentrates across layers, and implements multi-stage token reduction with a pyramid dropping ratio. FitPrune [45] formulates token pruning as a statistical distribution fitting problem, minimizing the divergence of both text-visual and intra-visual attention before and after pruning based on a small calibration set. Alternatively, diversity-based methods prioritize maximizing token diversity to reduce redundancy. DivPrune [3] formulates pruning as a Max-Min Diversity Problem [29, 31] to maximize minimum pairwise distances among selected tokens, and DART [39] selects tokens with the lowest similarity to pivot tokens. Nevertheless, all these methods perform static, irreversible pruning, potentially causing premature loss of tokens that become important in deeper layers.

3. Method

3.1. Preliminary

LVLMs are designed to process both text and visual inputs. For the text input, it is embedded into M textual tokens $X_T = [t_1, t_2, \ldots, t_M] \in \mathbb{R}^{M \times D}$ through the language model's embedding layer. For the visual input, a vision encoder extracts visual features, which are subsequently projected into N visual tokens $X_V = [v_1, v_2, \ldots, v_N] \in \mathbb{R}^{N \times D}$ via a projector layer, where typically $N \gg M$.

These text and visual tokens are concatenated as $X = [X_T, X_V] \in \mathbb{R}^{(M+N) \times D}$ and passed through the language model for autoregressive generation. At each step t, the model generates an output token y_t based on all previously generated tokens: $y_t \sim P(\cdot \mid y_{<t}, X_T, X_V)$, where $P(\cdot)$ denotes the conditional probability distribution from the language model, and D represents the hidden dimension.

3.2. ALVTS Framework

In this section, we present ALVTS, a novel framework that performs layer-wise dynamic token selection. At each layer of the language model, ALVTS computes importance scores for all visual tokens using a low-rank approximation [13] of the attention mechanism. Based on these scores, the most informative visual tokens are selected to participate in the layer's computation along with all text tokens, while the remaining visual tokens skip the layer. After processing, all tokens are reassembled to form the complete input sequence for the next layer. The pipeline is illustrated in Fig. 2.

Token Importance Estimation via Low-Rank Approximation. We employ a token selector $\mathcal{P}$ to compute importance scores for each visual token. To avoid the overhead of full attention computation, we design the selector using low-rank approximations of the layer's query and key projection matrices in the self-attention module:

$$\tilde{W}_Q = U_Q V_Q, \quad \tilde{W}_K = U_K V_K, \tag{1}$$

where $U_Q, U_K \in \mathbb{R}^{D\times R}$ and $V_Q, V_K \in \mathbb{R}^{R\times D}$, with $R \ll D$ being the rank. Given the input sequence $X = [X_T, X_V]$, we compute the approximate queries and keys:

$$\tilde{Q} = X\tilde{W}_Q^\top, \quad \tilde{K} = X\tilde{W}_K^\top, \tag{2}$$

and derive the attention weights between all tokens:

$$\tilde{A} = \text{softmax}\left(\frac{\tilde{Q}\tilde{K}^\top}{\sqrt{d_k}}\right) \in \mathbb{R}^{(M+N)\times(M+N)}. \tag{3}$$

We partition the attention matrix $\tilde{A}$ into two components: $\tilde{A}_{V2V} \in \mathbb{R}^{N\times N}$ representing visual-to-visual attention, and $\tilde{A}_{T2V} \in \mathbb{R}^{M\times N}$ representing text-to-visual attention. For the i-th visual token, we compute its visual-to-visual score as the average attention it receives from all visual tokens, and its text-to-visual score as the average attention it receives from all text tokens:

$$\mathcal{S}_{V2V}(i) = \frac{1}{N}\sum_{j=1}^{N} \tilde{A}_{V2V}(j,i), \tag{4}$$

$$\mathcal{S}_{T2V}(i) = \frac{1}{M}\sum_{j=1}^{M} \tilde{A}_{T2V}(j,i). \tag{5}$$

The final importance score is computed as the product of these two components:

$$\mathcal{S}(i) = \mathcal{S}_{V2V}(i) \cdot \mathcal{S}_{T2V}(i), \quad i = 1, 2, \ldots, N. \tag{6}$$

This multiplicative combination ensures that tokens are deemed important only when they are relevant to both the visual and textual contexts. Tokens with low scores in either dimension are considered less critical and can be safely skipped for the current layer.

Layer-wise Token Selection and Processing. Based on the computed importance scores, we perform top-k selection to identify the most informative visual tokens:

$$X_V^{(\text{select})} = \text{TopK}(X_V, \mathcal{S}, k), \tag{7}$$

where $X_V^{(\text{select})}$ contains k visual tokens and $k = \lfloor r \cdot N \rfloor$ with r being the selection ratio. The complementary set of skipped tokens is denoted as $X_V^{(\text{skip})}$, which contains the remaining $(N-k)$ visual tokens.

The selected tokens $X_V^{(\text{select})}$ together with all text tokens participate in the layer's computation, while $X_V^{(\text{skip})}$ skip the layer directly. At the output of layer ℓ, the processed tokens and skipped tokens are recombined to form the input for the next layer. Note that we maintain the original token ordering throughout this process to preserve positional information.

A key characteristic of ALVTS is that the token selection process is performed independently at each layer. Different layers may select different subsets of visual tokens based on their specific attention patterns, which enables adaptive and context-aware token compression throughout the model.

3.3. Token Selector Optimization

First, we initialize the low-rank projection matrices U_Q, V_Q, U_K, V_K using Singular Value Decomposition (SVD) of the original projection matrices [11]. Specifically, for the query projection $W_Q \in \mathbb{R}^{D\times D}$, we perform SVD to obtain $W_Q = U_{\text{full}}\Sigma V_{\text{full}}^\top$, where Σ contains the singular values in descending order. We retain the top R singular value components to construct the initial low-rank matrices U_Q and V_Q:

$$\Sigma_R^{1/2} = \text{diag}(\sqrt{\sigma_1}, \sqrt{\sigma_2}, \ldots, \sqrt{\sigma_R}) \in \mathbb{R}^{R\times R}, \tag{8}$$

$$U_Q = U_{\text{full}}[:, :R] \cdot \Sigma_R^{1/2} \in \mathbb{R}^{D\times R}, \tag{9}$$

$$V_Q = \Sigma_R^{1/2} \cdot V_{\text{full}}[:R, :]^\top \in \mathbb{R}^{R\times D}. \tag{10}$$

The same procedure is applied to initialize U_K and V_K from the original key projection W_K.

After initialization, we further adapt the selector to align its importance scoring behavior with that of the original model. Specifically, for each input sample, we derive the reference token importance scores $\mathcal{S}^*$ from the original model's attention patterns and align our low-rank approximation $\mathcal{S}$ by minimizing the reconstruction error:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{S}(i) - \mathcal{S}^*(i))^2. \tag{11}$$

Each layer's selector is optimized independently, and we only update the parameters of the selectors, while keeping the original LVLM frozen. This design is highly efficient.

Table 1. Performance comparison of LLaVA-1.5-7B under different token reduction ratios. "Avg." refers to the average performance across 8 benchmarks. Best results are shown in **bold**.

Method	AI2D	POPE	VQA^{Text}	OKVQA	VizWiz	COCO	NoCaps	RealWorld	Avg.
Upper Bound, 576 Tokens (100%)									
LLaVA-1.5-7B	55.21	85.84	58.18	53.42	54.32	110.52	105.61	56.08	100.00%
Retain 192 Tokens (↓ 67%)									
FastV *(ECCV-24)*	54.73	77.79	57.91	51.77	54.95	107.49	102.26	53.33	97.07%
VTW *(AAAI-25)*	54.99	74.86	50.60	30.60	50.66	39.85	26.66	50.46	71.95%
PDrop *(CVPR-25)*	54.40	84.67	57.11	**52.37**	54.42	109.28	103.76	53.20	98.19%
DART *(EMNLP-25)*	54.73	82.45	57.40	51.50	54.95	108.93	103.84	54.25	98.13%
ALVTS (*Ours*)	**55.44**	**85.72**	**58.29**	52.12	**55.05**	**111.68**	**104.50**	**54.64**	**99.60%**
Retain 128 Tokens (↓ 78%)									
FastV *(ECCV-24)*	54.02	72.10	56.95	49.90	55.14	101.99	97.06	51.11	93.75%
VTW *(AAAI-25)*	53.04	41.25	44.26	22.87	50.76	9.50	5.76	44.05	56.13%
PDrop *(CVPR-25)*	53.92	81.82	56.18	50.27	52.35	107.86	97.01	51.37	95.13%
DART *(EMNLP-25)*	54.37	80.26	56.64	51.17	**55.28**	107.80	**102.86**	**53.99**	97.26%
ALVTS (*Ours*)	**54.57**	**85.99**	**57.92**	**52.13**	54.90	**110.32**	102.30	**53.99**	**98.77%**
Retain 64 Tokens (↓ 89%)									
FastV *(ECCV-24)*	53.56	59.55	55.32	45.14	54.91	82.62	79.01	47.32	85.13%
VTW *(AAAI-25)*	50.26	31.67	42.85	19.96	50.34	9.80	6.06	43.53	52.98%
PDrop *(CVPR-25)*	53.30	72.39	53.36	45.10	51.56	99.26	85.45	47.97	88.52%
DART *(EMNLP-25)*	53.59	74.02	54.53	47.99	55.46	103.89	**97.70**	50.85	93.27%
ALVTS (*Ours*)	**54.63**	**83.87**	**57.41**	**49.66**	**55.65**	**105.89**	97.60	**53.20**	**96.73%**

Taking LLaVA-1.5-7B as an example, the entire optimization process can be completed within 15 minutes. Furthermore, our token selector introduces negligible parameter overhead to the base model. The detailed rank settings, parameter counts, and overhead for different models are presented in Tab. 4.

4. Experiments

4.1. Experimental Setup

Models. We evaluate ALVTS on three representative LVLMs: LLaVA-1.5 (7B and 13B variants) [23], LLaVA-NeXT-7B [24], and Qwen2.5-VL-3B-Instruct [5]. LLaVA-1.5 employs a fixed-length visual token sequence of 576 tokens. LLaVA-NeXT-7B supports dynamic image resolutions with variable-length visual tokens, enabling more flexible visual processing. Qwen2.5-VL-3B-Instruct supports naive dynamic resolution, generating variable-length token sequences without resizing images to a fixed input size. These models provide comprehensive evaluation scenarios for our compression method.

Benchmarks. We conduct experiments on 8 multimodal benchmarks: AI2D [17], POPE [18], TextVQA [34], OKVQA [26], VizWiz [15], COCO Caption [20], NoCaps [1], and RealWorldQA. These benchmarks cover diverse tasks including general VQA, text-based VQA, image captioning, and multimodal reasoning. Further details are presented in the Appendix.

Table 2. Performance comparison of LLaVA-1.5-13B. We report the average performance across 8 benchmarks, with detailed per-benchmark results provided in the Appendix.

Method	Ratio	Avg.	Ratio	Avg.	Ratio	Avg.
FastV		98.71%		96.41%		91.20%
VTW		85.19%		69.16%		48.35%
PDrop	67%	97.88%	78%	96.56%	89%	92.66%
DART		99.16%		97.41%		93.98%
ALVTS		**99.86%**		**99.12%**		**96.63%**

Comparison Methods. We compare our method with 4 popular visual token compression methods. FastV [9] prunes visual tokens with low attention scores after the second layer of the LLM. VTW [21] strategically withdraws vision tokens at a certain layer, enabling only text tokens to engage in the computation of subsequent layers. PyramidDrop [40] progressively drops visual tokens across stages as token redundancy increases in deeper layers. DART [39]

Table 3. Performance Comparison of LLaVA-NeXT-7B. The vanilla number of visual tokens varies by dataset due to dynamic image processing (up to 2880 tokens). "Avg." refers to the average performance across 8 benchmarks. Best results are shown in **bold**.

Method	AI2D	POPE	VQA^{Text}	OKVQA	VizWiz	COCO	NoCaps	RealWorld	Avg.
Upper Bound, 2880 Tokens (100%)									
LLaVA-NeXT-7B	65.32	86.41	61.39	44.23	60.62	99.83	88.28	57.91	100.0%
Retain 320 Tokens (↓ 89%)									
FastV *(ECCV-24)*	62.79	74.47	57.53	38.91	57.60	82.56	74.54	52.94	89.70%
VTW *(AAAI-25)*	60.75	0.27	37.88	9.61	52.10	5.97	3.11	42.22	43.14%
PDrop *(CVPR-25)*	62.60	81.80	56.93	**43.27**	58.60	**94.68**	79.53	54.25	94.54%
DART *(EMNLP-25)*	64.48	81.58	57.02	40.37	58.79	91.02	78.89	55.56	93.84%
ALVTS (*Ours*)	**64.51**	**87.09**	**58.37**	41.70	**59.36**	94.37	**80.24**	**56.60**	**96.25%**

Table 4. Token selector rank settings, parameter counts, and parameter overhead per decoder layer for different models.

Model	Rank	Params/Layer	Overhead/Layer
LLaVA-1.5-7B	256	4.2M	2.0%
LLaVA-NeXT-7B	256	4.2M	2.0%
LLaVA-1.5-13B	256	5.2M	1.7%
Qwen2.5-VL-3B	128	0.8M	1.1%

prunes tokens based on their duplication with pivot tokens, which guarantees maximal information diversity. To ensure a fair comparison, we adjust the hyperparameters of each method to match the FLOPs of FastV, which performs visual token pruning once after the second layer of the LLM.

Implementation Details. To optimize the token selectors, we randomly select 256 samples from the LLaVA-655k dataset [22]. To keep parameter overhead below 2% per decoder layer, we set the rank of low-rank weight matrices to 256 for LLaVA-1.5-7B and LLaVA-NeXT-7B, and to 128 for Qwen2.5-VL-3B-Instruct (see Tab. 4). For performance evaluation on test benchmarks, we employ the LMMs-Eval evaluation framework [48] across all the baselines and models.

4.2. Main Results

Tab. 1 presents comprehensive evaluation results of ALVTS against existing approaches across 8 benchmarks under three different token reduction ratios. At a 67% reduction ratio, ALVTS achieves an average performance of 99.60%, surpassing all baseline methods, with a 1.41% margin over the second-best method, PyramidDrop. At a more aggressive 89% reduction ratio, our method maintains a 96.73% average performance, consistently outperforming all baselines by 3.46% over DART, 8.21% over PyramidDrop, and 11.60% over FastV. The maintained high performance on VQA^{Text} (57.41) and VizWiz (55.65) at this extreme compression setting suggests that our dynamic token selection strategy effectively preserves text-relevant visual features and adapts well to real-world, challenging images.

Table 5. Performance Comparison on Qwen2.5-VL-3B.

Method	AI2D	POPE	COCO	NoCaps	RealWorld	Avg.
Upper Bound, All Tokens (100%)						
Vanilla	78.72	86.91	99.78	106.46	59.35	100.0%
Token Reduction (↓ 89%)						
FastV	68.26	73.98	72.18	77.78	49.80	80.23%
PDrop	67.03	67.84	73.54	84.81	44.84	78.43%
ALVTS	**68.39**	**86.58**	**78.23**	**85.72**	**51.37**	**86.39%**

To validate the scalability of our approach, we further conduct experiments on the larger LLaVA-1.5-13B model. As shown in Tab. 2, ALVTS consistently achieves superior performance across all compression ratios. At a 67% reduction ratio, our method reaches an average performance of 99.86%. When the compression ratio increases to 78% and 89%, ALVTS maintains 99.12% and 96.63% average performance respectively, outperforming the second best method by 1.71% and 2.65% at these settings. Detailed comparison results on LLaVA-1.5-13B across 8 benchmarks are presented in the Appendix.

4.3. ALVTS with Higher Resolution

We also present evaluations on LLaVA-NeXT-7B, a high-resolution LVLM that processes significantly more visual tokens. As shown in Tab. 3, when retaining only 320 tokens (from up to 2880 input tokens), ALVTS preserves 96.25% of the original model's average performance, outperforming PyramidDrop and DART by 1.7% and 2.4% respectively. The advantage is particularly pronounced on the POPE benchmark, where ALVTS achieves 87.09, even slightly surpassing the vanilla model's performance. These results demonstrate the effectiveness of ALVTS for process-

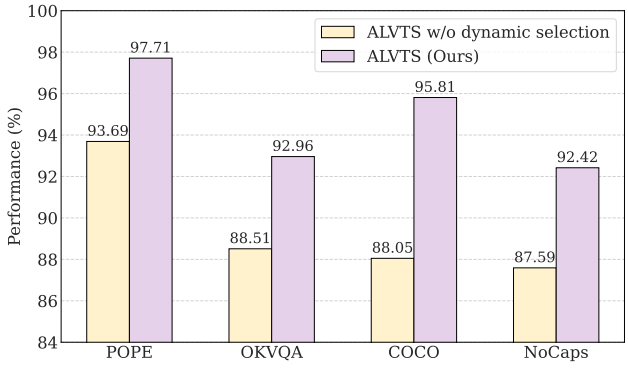


Figure 3. Ablation study of dynamic token selection mechanism.

Table 6. Efficiency comparison on POPE dataset. Experiments are on LLaVA-1.5-7B with a single 4090 GPU.

Method	Time ↓	Latency ↓	Prefill ↓	Acc. ↑
LLaVA-1.5-7B	31:44	211ms	165ms	100%
FastV (↓ 60%)	26:51	179ms	126ms	93.2%
ALVTS (↓ 89%)	**23:28**	**156ms**	**103ms**	**97.7%**

ing high-resolution visual input.

4.4. ALVTS with Qwen Architecture

To verify the generalization of ALVTS beyond LLaVA-based models, we conduct experiments on Qwen2.5-VL-3B, with results shown in Tab. 5. At an 89% token reduction rate, ALVTS retains 86.39% of the original performance on average, outperforming FastV (80.23%) and PyramidDrop (78.43%) by 6.16 and 7.96 percentage points, respectively. These results confirm that ALVTS effectively generalizes across different vision-language model architectures.

4.5. Efficiency Analysis

To evaluate the efficiency of ALVTS, we compare total inference time, end-to-end latency, prefill time, and accuracy on LLaVA-1.5-7B. As shown in Tab. 6, under an 89% token reduction ratio, ALVTS maintains a 97.7% accuracy. Compared to the vanilla model, ALVTS decreases the average per-sample latency from 211ms to 156ms (1.35× speedup), and reduces the prefill time from 165ms to 103ms (1.6× speedup). When compared to FastV with a 60% token reduction ratio, ALVTS delivers comparable inference speed while achieving 4.5 percentage points higher accuracy (97.7% vs. 93.2%). These results demonstrate that ALVTS attains a better trade-off between inference efficiency and model performance.

4.6. Ablation Studies

Effectiveness of Layer-wise Dynamic Token Selection. To validate the effectiveness of the dynamic token selection mechanism, we conduct an ablation study by comparing ALVTS with a static pruning baseline (ALVTS w/o dynamic selection) on LLaVA-1.5-7B with 64 visual tokens retained. The baseline performs one-time token pruning after the second layer and maintains this pruned token set throughout all subsequent layers, following the same strategy as FastV. As illustrated in Fig. 3, the full ALVTS method consistently outperforms the static pruning baseline. The dynamic selection mechanism brings notable improvements, especially on COCO and NoCaps, achieving gains of 7.76% and 4.83%, respectively. These results confirm that different decoder layers exhibit varying optimal token subsets, and static pruning leads to premature information loss that compromises model performance. By allowing less important visual tokens to skip certain layers while remaining accessible for subsequent processing, ALVTS effectively balances efficiency and performance, thereby validating the necessity of the dynamic selection mechanism.

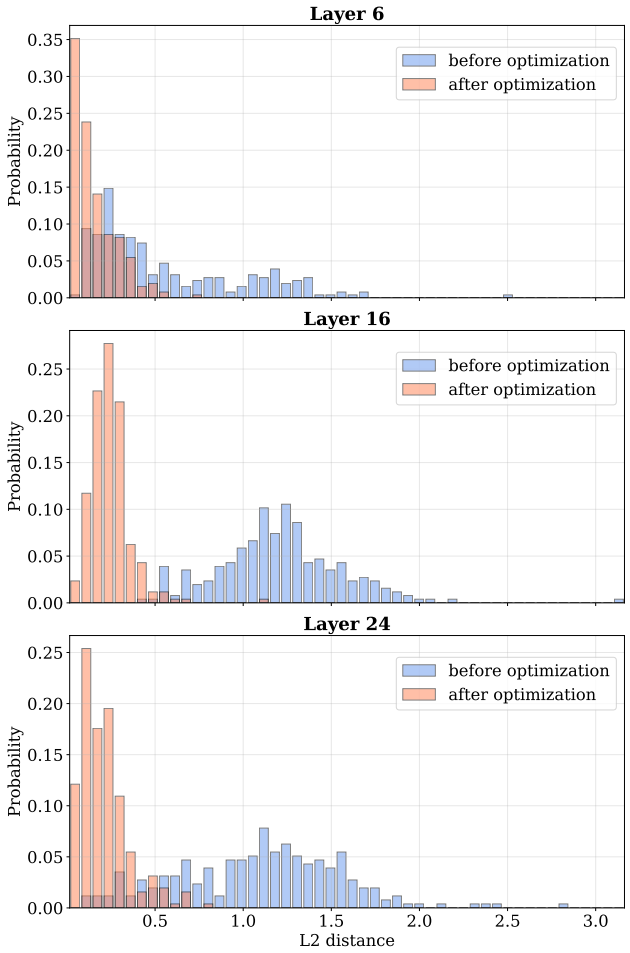


Figure 4. The distribution of L_2 distance between the approximate importance scores and the oracle importance scores.

Effectiveness of Token Selector Optimization. To val-

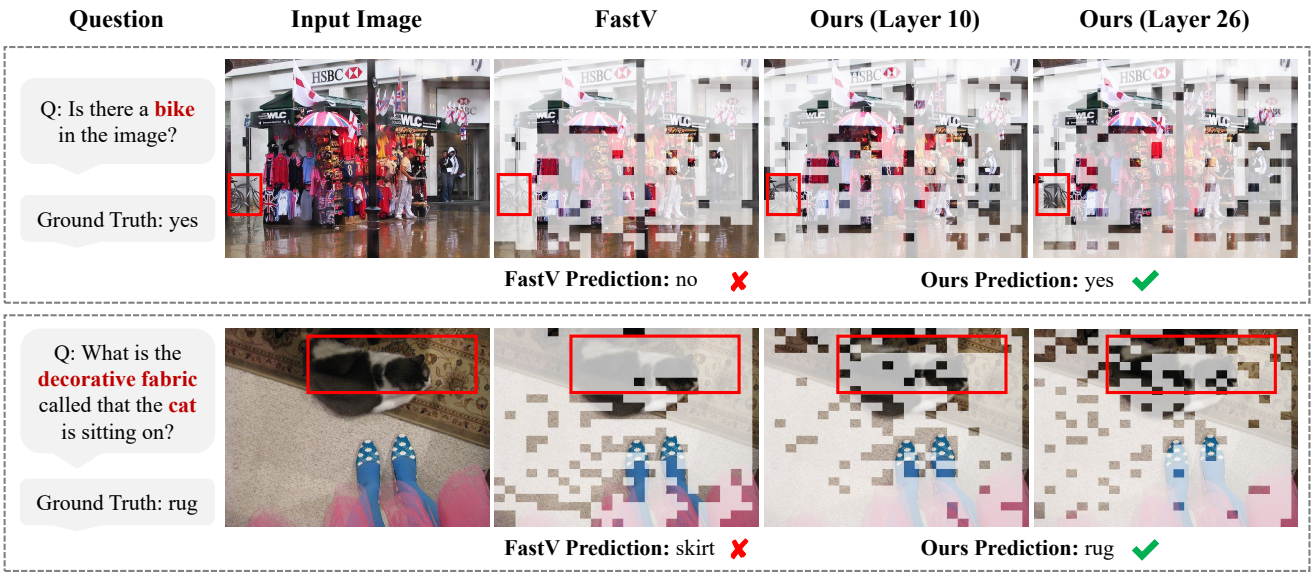


Figure 5. The case comparison between FastV and ALVTS. It presents original images alongside their pruned versions. Grey mask areas indicate discarded tokens. Red bounding boxes highlight key objects referenced in prompts.

idate the effectiveness of our low-rank token selector design, we evaluate how well it approximates the importance scoring behavior of the full attention mechanism. We randomly sample 1,000 instances from the AI2D dataset and measure the L_2 distance between the approximate importance scores (derived from our low-rank selector) and the oracle importance scores (computed via full-rank query and key projections) of visual tokens. Fig. 4 presents the L_2 distance distributions at several layers before and after optimization. The blue histograms depict the distance distribution obtained using SVD-initialized weights, while the orange histograms show the results after optimization with the objective defined in Eq. (11). The results reveal a pronounced improvement in approximation fidelity: the L_2 distance distributions become significantly more concentrated near zero after optimization, with the peak probability density shifting closer to the origin. This demonstrates that our low-rank token selector accurately captures the importance scoring patterns inherent in the full attention mechanism, thereby ensuring reliable token selection.

4.7. Visualizations Results

To provide a deeper understanding of our ALVTS framework, we visualize the selected visual tokens in Fig. 5. We present two examples where our method yields correct predictions while FastV fails. The figure illustrates the token selection results from two ALVTS layers alongside FastV's static pruning result. For FastV, the pruning layer fails to attend to the visual regions mentioned in the prompts (e.g., "bike" and "decorative fabric"), thus leading to incorrect answers. In contrast, ALVTS exhibits distinct visual region preferences across different layers. Even when shallow layers do not initially focus on the target objects, deeper layers are able to attend to and process these critical regions, ultimately ensuring accurate predictions. For instance, as shown in the bottom row, while FastV discards the tokens corresponding to the "decorative fabric," ALVTS correctly identifies these critical regions at layer 10 and layer 26. This highlights the key advantage of ALVTS: by preserving token accessibility across layers and enabling dynamic token selection based on each layer's attention pattern, our method maintains robust visual grounding capabilities throughout inference.

5. Conclusion

In this paper, we propose ALVTS, a novel framework that challenges the conventional paradigm of static token pruning in LVLMs. Unlike existing methods that permanently discard tokens at a specific layer, ALVTS performs adaptive layer-wise visual token selection. Each layer can process the tokens it deems important while keeping all tokens accessible throughout the model. To achieve this, we design a token selector based on importance consistency constrained low-rank approximation, which faithfully emulates the full attention mechanism in token selection behavior without model retraining. Comprehensive experiments demonstrate that ALVTS achieves superior efficiency-accuracy trade-offs, making it a practical solution for deploying LVLMs in real-world settings. By respecting the heterogeneous visual information needs across layers, we can indeed turn one layer's trash into another layer's treasure.

References

[1] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8948–8957, 2019. 5

[2] Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, 2023. 2

[3] Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9392–9401, 2025. 3

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 1

[5] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1, 2, 5

[6] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024. 1

[7] Jun Chen, Han Guo, Kai Yi, Boyang Li, and Mohamed Elhoseiny. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18030–18040, 2022. 1

[8] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer, 2024. 1

[9] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024. 1, 3, 5

[10] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 2

[11] James W Demmel. *Applied numerical linear algebra*. SIAM, 1997. 4

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

[13] Gene H Golub, Alan Hoffman, and Gilbert W Stewart. A generalization of the eckart-young-mirsky matrix approximation theorem. *Linear Algebra and its applications*, 88: 317–327, 1987. 4

[14] 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. 1

[15] 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. 5

[16] Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. On the computational complexity of self-attention. In *International conference on algorithmic learning theory*, pages 597–619. PMLR, 2023. 2

[17] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pages 235–251. Springer, 2016. 5

[18] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 5

[19] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5971–5984, 2024. 1

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[21] Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5334–5342, 2025. 1, 5

[22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 6

[23] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024. 2, 5

[24] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024. 1, 2, 5

[25] Chuofan Ma, Yi Jiang, Jiannan Wu, Zehuan Yuan, and Xiaojuan Qi. Groma: Localized visual tokenization for grounding multimodal large language models. In *European Conference on Computer Vision*, pages 417–435. Springer, 2024. 1

[26] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings*

of the IEEE/cvf conference on computer vision and pattern recognition, pages 3195–3204, 2019. 1, 5

[27] Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. Improving multimodal datasets with image captioning. *Advances in neural information processing systems*, 36:22047–22069, 2023. 1

[28] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 1

[29] Daniel Cosmin Porumbel, Jin-Kao Hao, and Fred Glover. A simple and effective algorithm for the maxmin diversity problem. *Annals of Operations Research*, 186(1):275–293, 2011. 3

[30] 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. 2

[31] Mauricio GC Resende, Rafael Martí, Micael Gallego, and Abraham Duarte. Grasp and path relinking for the max–min diversity problem. *Computers & Operations Research*, 37 (3):498–508, 2010. 3

[32] Kele Shao, Keda Tao, Kejia Zhang, Sicheng Feng, Mu Cai, Yuzhang Shang, Haoxuan You, Can Qin, Yang Sui, and Huan Wang. When tokens talk too much: A survey of multimodal long-context token compression across images, videos, and audios. *arXiv preprint arXiv:2507.20198*, 2025. 1, 3

[33] Ahmed Sharshar, Latif U Khan, Waseem Ullah, and Mohsen Guizani. Vision-language models for edge networks: A comprehensive survey. *IEEE Internet of Things Journal*, 2025. 1

[34] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 5

[35] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[37] Ao Wang, Fengyuan Sun, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. [cls] token tells everything needed for training-free efficient mllms. *arXiv preprint arXiv:2412.05819*, 2024. 1, 3

[38] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. 1, 2

[39] Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9972–9991, 2025. 3, 5

[40] 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. 3, 5

[41] Yunqiu Xu, Linchao Zhu, and Yi Yang. Mc-bench: A benchmark for multi-context visual grounding in the era of mllms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17675–17687, 2025. 1

[42] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 1

[43] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19792–19802, 2025. 1, 3

[44] 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

[45] Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 22128–22136, 2025. 1, 3

[46] Hao Yin, Guangzong Si, and Zilei Wang. Lifting the veil on visual information flow in mllms: Unlocking pathways to faster inference. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9382–9391, 2025. 1

[47] Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Yaqi Xie, Katia Sycara, Haitao Mi, and Dong Yu. Vscan: Rethinking visual token reduction for efficient large vision-language models. *arXiv preprint arXiv:2505.22654*, 2025. 1

[48] 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. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 881–916, 2025. 6

[49] Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20857–20867, 2025. 1, 3

[50] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 1