# CENTROIDPRUNE: CENTROID MATCHING FOR VISUAL TOKEN PRUNING IN VISION-LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Vision-Language Models (VLMs) incur substantial inference cost because a single image is converted into a large number of visual tokens. Token pruning addresses this by removing less informative visual tokens. Existing methods rank tokens with an importance measure such as attention score or token similarity, and retain the top-ranked ones. However, these measures do not reflect how removing tokens changes the model's output, and existing methods degrade sharply as the pruning ratio increases. In this work, we analyze the effect of pruning on the attention output and show that the induced change decomposes into two factors: the removed attention mass and the directional deviation of the removed tokens. Attention-based pruning minimizes the former while leaving the latter uncontrolled, and therefore does not minimize the change in the attention output. Motivated by this analysis, we propose CentroidPrune, a training-free visual token pruning method. We formulate token pruning as a subset selection problem that directly minimizes the distance between the original attention output and the output after pruning and renormalization. Extensive experiments on various VLMs and multimodal benchmarks show that CentroidPrune consistently outperforms prior methods, retaining $98.4\%$ of the unpruned performance while discarding up to $94.4\%$ of the visual tokens.

## 1 INTRODUCTION

Vision-Language Models (VLMs) have demonstrated remarkable capabilities in multimodal understanding by integrating the reasoning capabilities of Large Language Models (LLMs) with visual inputs (Liu et al., 2023; Dai et al., 2023; Liu et al., 2024a; Chen et al., 2024b; Bai et al., 2025). Despite their impressive progress, the substantial computational and memory costs of VLMs remain a major challenge. A primary source of this overhead is the large number of visual tokens extracted from visual inputs, which substantially increase the sequence length processed by the LLM. To address this challenge, token pruning has emerged as a practical solution that selectively removes redundant or less informative visual tokens while preserving model performance (Chen et al., 2024a; Yang et al., 2025; Alvar et al., 2025).

Existing token pruning methods can be broadly categorized into two groups based on where pruning is performed. The first performs pruning before visual tokens are fed into the base LLM (Shang et al., 2025; Yang et al., 2025; Arif et al., 2025; Kim et al., 2026). The second performs pruning within the LLM after processing several layers (Chen et al., 2024a; Zhang et al., 2024; Xing et al., 2024). Both approaches compute an importance measure, such as token similarity or attention score, and retain the top-ranked tokens accordingly. However, these criteria do not directly reflect how removing tokens changes the model's output. Consequently, existing methods suffer from substantial performance degradation as more tokens are removed.

In this work, we analyze how visual token pruning changes the attention output. We show that the change induced by removing a set of visual tokens decomposes exactly into two factors: the removed attention mass and the directional deviation of the removed tokens. This decomposition reveals a limitation of existing attention-based pruning methods. Retaining the top-$k$ tokens with the highest attention scores minimizes the removed attention mass, but leaves the directional deviation



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

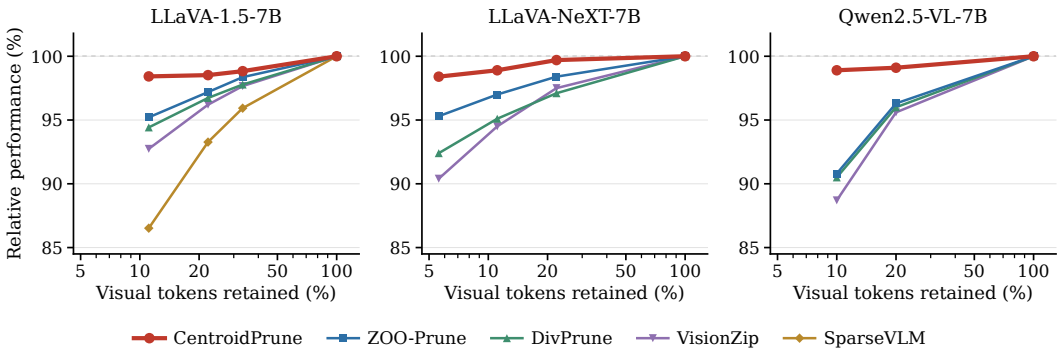


Figure 1: Relative performance under increasing pruning ratios. Each panel corresponds to one backbone. The horizontal axis is the percentage of visual tokens retained, and the vertical axis is the average performance over all benchmarks, normalized by the unpruned model.

uncontrolled. Minimizing the removed attention mass alone therefore does not minimize the change in the attention output.

Building on this insight, we introduce CentroidPrune, a training-free visual token pruning method. Instead of estimating token importance using a proxy metric, our method directly minimizes the change in the attention output caused by pruning. Specifically, we formulate visual token pruning as a subset selection problem that seeks to minimize the distance between the original attention output and the output after pruning and renormalization. CentroidPrune then identifies a subset of visual tokens that best preserves the original attention output, enabling aggressive visual token reduction while maintaining performance close to that of the original VLM.

As shown in Figure 1, CentroidPrune remains nearly flat as visual tokens are removed, while existing methods degrade sharply under the same budgets. On LLaVA-1.5-7B, it retains $98.42\%$ of the unpruned model with only $64$ visual tokens, already surpassing every baseline evaluated with three times that budget. On LLaVA-NeXT-7B, it retains $98.4\%$ while discarding $94.4\%$ of the visual tokens, matching the average that the strongest baseline attains with four times the budget.

Our contributions are summarized as follows:

- We show that the change in the attention output caused by visual token pruning decomposes exactly into a removed attention mass and a directional deviation, and that attention-based pruning controls only the former.
- We propose CentroidPrune, a training-free method that formulates token pruning as a subset selection problem directly minimizing the change in the attention output, requiring no additional forward pass.
- CentroidPrune achieves state-of-the-art performance across multiple VLMs and benchmarks, with degradation remaining small even under aggressive pruning ratios.

## 2 Related Work

**Vision-Language Models.** Vision-Language Models (VLMs) extend the reasoning and instruction-following capabilities of Large Language Models (LLMs) to visual inputs (Liu et al., 2023; Dai et al., 2023; Liu et al., 2024a; Chen et al., 2024b; Bai et al., 2025). A common design encodes an image with a pretrained vision encoder such as CLIP (Radford et al., 2021) and maps the resulting features into the language embedding space through a lightweight projector (Liu et al., 2023; Li et al., 2023b). The projected features are then fed to the LLM decoder as a sequence of visual tokens together with the text tokens. This sequential visual representation allows VLMs to inherit the architecture and pretrained weights of LLMs without modification, and has become the dominant paradigm across a wide range of multimodal tasks. However, it also substantially increases the total token count. LLaVA (Liu et al., 2023; 2024a;b) typically encodes $336\times336$ images into

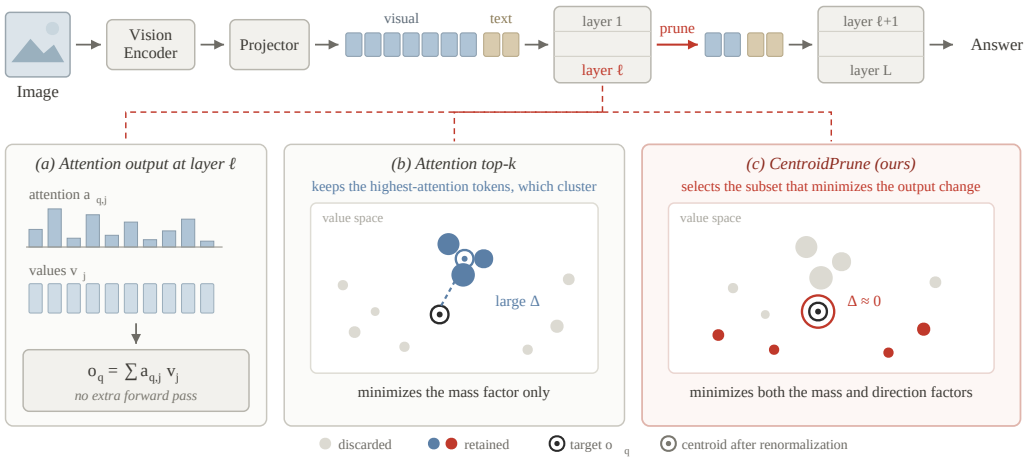


Figure 2: **Overview of CentroidPrune.** At decoder layer $\ell$, the attention weights and value vectors of the last query position give the attention output $\mathbf{o}_q$ (a), which requires no additional forward pass. Attention top-$k$ keeps the tokens carrying the most attention mass, which tend to cluster, so the renormalized centroid moves away from $\mathbf{o}_q$ (b). CentroidPrune instead selects the subset whose renormalized centroid reproduces $\mathbf{o}_q$ (c).

576 visual tokens, and produces up to 2,880 tokens at 672×672 resolution (Li et al., 2024; Guo et al., 2024). Since the inference cost of the LLM grows with the input sequence length, the number of visual tokens directly determines the computational and memory overhead of VLMs (Vaswani et al., 2017).

**Visual Token Pruning.** Visual token pruning reduces this overhead by removing redundant or less informative visual tokens at inference time, and existing methods can be grouped by where pruning is performed. The first group prunes before the visual tokens are fed into the base LLM. These methods rely on signals available outside the LLM, such as the `[CLS]` attention of the vision encoder (Shang et al., 2025; Yang et al., 2025; Arif et al., 2025), token merging based on feature similarity (Bolya et al., 2022), or sensitivity estimated at the projector output (Kim et al., 2026). Since token importance is assessed individually, attention-based scores tend to retain tokens that are similar to each other. DivPrune (Alvar et al., 2025) instead operates at the set level and formulates token pruning as a Max-Min Diversity Problem to maximize the minimum pairwise distance among the selected tokens. The second group prunes within the LLM, using the self-attention computed in the decoder layers so that the selection is guided by the text tokens. FastV (Chen et al., 2024a) ranks visual tokens by the attention they receive at a shallow layer. SparseVLM (Zhang et al., 2024) selects visual-relevant text tokens to rate the significance of visual tokens and adaptively determines the sparsification ratio of each layer. PyramidDrop (Xing et al., 2024) partitions the forward pass into several stages and drops a predefined ratio of image tokens at the end of each stage.

Despite differing in where the signal is obtained, both groups estimate token importance through a proxy metric and retain the top-ranked tokens. In contrast, we analyze how the attention output itself changes once tokens are removed and the remaining attention weights are renormalized, and select the subset that directly minimizes this change.

## 3 Method

In this section, we present CentroidPrune, a training-free visual token pruning method. We first introduce the notation and describe how pruning affects the attention output (Section 3.1). We then derive an exact decomposition of this effect and show that attention-based pruning controls only one of its two factors (Section 3.2). Building on this analysis, we present our selection criterion and the procedure that solves it (Section 3.3). Figure 2 shows an overview of our method.

### 3.1 PRELIMINARY

**Vision-Language Model.** Given an image, a VLM encodes it with a pretrained vision encoder and maps the resulting features into the language embedding space through a projector, producing a set of visual tokens indexed by $\mathcal{V} = \{1, \ldots, N\}$. These tokens are concatenated with the text tokens, indexed by $\mathcal{T}$, and the combined sequence is processed by the LLM decoder. Visual token pruning selects a subset $\mathcal{S} \subseteq \mathcal{V}$ with $|\mathcal{S}| = k \ll N$ and discards the remaining tokens, so that the subsequent decoder layers operate on a shorter sequence. Following common practice, we retain all text tokens, since $|\mathcal{T}|$ is negligible compared to $N$.

**Attention in VLM Decoders.** Let $\ell$ denote a decoder layer, $h$ an attention head, and $q$ a query position. We omit the head index unless it is needed. The attention output at $q$ is the attention-weighted average of the value vectors over all tokens,

$$\mathbf{o}_q = \sum_{j \in \mathcal{V} \cup \mathcal{T}} a_{q,j}\, \mathbf{v}_j, \qquad \sum_{j \in \mathcal{V} \cup \mathcal{T}} a_{q,j} = 1, \tag{1}$$

where $a_{q,j}$ is the self-attention weight assigned to token $j$ and $\mathbf{v}_j$ is its value vector at layer $\ell$. Since text tokens are never pruned, we denote their contribution and their attention mass by

$$\mathbf{c}_q = \sum_{j \in \mathcal{T}} a_{q,j}\, \mathbf{v}_j, \qquad \tau_q = \sum_{j \in \mathcal{T}} a_{q,j}, \tag{2}$$

both of which remain constant under pruning. Note that $\mathbf{o}_q$ is already computed during the forward pass of layer $\ell$ and thus requires no additional computation.

**Effect of Pruning.** When the tokens outside $\mathcal{S}$ are removed, the softmax is renormalized over the retained tokens $\mathcal{S} \cup \mathcal{T}$, and the attention output at $q$ becomes

$$\mathbf{o}_q(\mathcal{S}) = \frac{\mathbf{c}_q + \sum_{j \in \mathcal{S}} a_{q,j}\, \mathbf{v}_j}{\tau_q + \sum_{j \in \mathcal{S}} a_{q,j}}. \tag{3}$$

Pruning therefore does not merely delete the contribution of the removed tokens. Because of the renormalization, the attention output is replaced by a reweighted average over the retained tokens, where the retained weights are scaled up by a factor determined by how much attention mass was removed. Consequently, $\mathbf{o}_q(\mathcal{S})$ can differ from $\mathbf{o}_q$ even when the removed tokens individually carried small attention weights.

### 3.2 ANALYSIS: MASS AND DIRECTION

Let $\mathcal{R} = \mathcal{V} \setminus \mathcal{S}$ denote the set of removed visual tokens. We define the removed attention mass $m_{\mathcal{R}}$ and the attention output restricted to the removed tokens $\bar{\mathbf{v}}_{\mathcal{R}}$ as

$$m_{\mathcal{R}} = \sum_{j \in \mathcal{R}} a_{q,j}, \qquad \bar{\mathbf{v}}_{\mathcal{R}} = \frac{1}{m_{\mathcal{R}}} \sum_{j \in \mathcal{R}} a_{q,j}\, \mathbf{v}_j, \tag{4}$$

where $\bar{\mathbf{v}}_{\mathcal{R}}$ takes the same form as $\mathbf{o}_q$ in Equation 1 but is computed over $\mathcal{R}$ alone. Since the attention weights sum to one over $\mathcal{V} \cup \mathcal{T}$, the denominator of Equation 3 satisfies $\tau_q + \sum_{j \in \mathcal{S}} a_{q,j} = 1 - m_{\mathcal{R}}$, while its numerator equals $\mathbf{o}_q - m_{\mathcal{R}} \bar{\mathbf{v}}_{\mathcal{R}}$. Substituting both into Equation 3 and subtracting $\mathbf{o}_q$ yields

$$\boldsymbol{\Delta}(\mathcal{S}) \;=\; \mathbf{o}_q(\mathcal{S}) - \mathbf{o}_q \;=\; \underbrace{\frac{m_{\mathcal{R}}}{1 - m_{\mathcal{R}}}}_{\text{mass factor}} \cdot \underbrace{\left(\mathbf{o}_q - \bar{\mathbf{v}}_{\mathcal{R}}\right)}_{\text{direction factor}}. \tag{5}$$

Equation 5 holds exactly for any subset $\mathcal{S}$, without any first-order expansion. The change induced by pruning is thus the product of a scalar determined by how much attention mass is removed, and a vector measuring how far the removed tokens deviate from the original attention output.

Since Equation 5 is a product, minimizing one factor alone does not guarantee that $\|\boldsymbol{\Delta}(\mathcal{S})\|$ is small. Existing attention-based methods retain the top-$k$ tokens with the largest $a_{q,j}$ (Chen et al., 2024a; Xing et al., 2024), which minimizes $m_{\mathcal{R}}$ and hence the mass factor, but leaves the direction factor unconstrained. Tokens with large attention weights are often concentrated in similar regions of the image (Alvar et al., 2025; Shang et al., 2025), so discarding everything else can place $\bar{\mathbf{v}}_{\mathcal{R}}$ far from $\mathbf{o}_q$. This motivates a selection criterion that targets $\|\boldsymbol{\Delta}(\mathcal{S})\|$ directly rather than through a proxy metric.

**Algorithm 1** CentroidPrune

**Input:** Attention weights $a$ and value vectors $\mathbf{v}$ at decoder layer $\ell$, number of tokens to retain $k$, number of rounds $T$
**Output:** Selected token indices $\mathcal{S}$
Attention output: $\mathbf{o}_q = \sum_{j \in \mathcal{V} \cup \mathcal{T}} a_{q,j} \mathbf{v}_j$
Text contribution: $\mathbf{c}_q = \sum_{j \in \mathcal{T}} a_{q,j} \mathbf{v}_j, \quad \tau_q = \sum_{j \in \mathcal{T}} a_{q,j}$
Initialize $\mathcal{S} \leftarrow \{\arg\max_{j \in \mathcal{V}} a_{q,j}\}$
Initialize $\mathbf{n} \leftarrow \mathbf{c}_q + \sum_{j \in \mathcal{S}} a_{q,j} \mathbf{v}_j, \quad d \leftarrow \tau_q + \sum_{j \in \mathcal{S}} a_{q,j}$
Round size: $B \leftarrow \lceil (k-1)/T \rceil$
**while** $|\mathcal{S}| < k$ **do**
    Candidate output: $\mathbf{o}_j = (\mathbf{n} + a_{q,j} \mathbf{v}_j)/(d + a_{q,j})$ for all $j \notin \mathcal{S}$
    Centroid error: $s_j = \sum_h \|\mathbf{o}_j^{(h)} - \mathbf{o}_q^{(h)}\|^2 / \|\mathbf{o}_q^{(h)}\|^2$
    Select $\mathcal{B}$ as the indices of the $B'$ smallest $s_j$, where $B' = \min(B, k - |\mathcal{S}|)$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{B}$
    Update $\mathbf{n} \leftarrow \mathbf{n} + \sum_{j \in \mathcal{B}} a_{q,j} \mathbf{v}_j, \quad d \leftarrow d + \sum_{j \in \mathcal{B}} a_{q,j}$
**end while**
**return** $\mathrm{sort}(\mathcal{S})$

### 3.3 CentroidPrune

Following the analysis in Section 3.2, we formulate visual token pruning as a subset selection problem that directly minimizes $\|\boldsymbol{\Delta}(\mathcal{S})\|$. From Equation 3, the attention output after pruning is a weighted average of the value vectors over the retained tokens, that is, their centroid in the value space. Preserving the attention output therefore amounts to selecting a subset whose centroid reproduces $\mathbf{o}_q$, which gives the objective

$$\mathcal{S}^* = \underset{\mathcal{S} \subseteq \mathcal{V},\ |\mathcal{S}| = k}{\arg\min} \sum_h \frac{\left\| \mathbf{o}_q^{(h)}(\mathcal{S}) - \mathbf{o}_q^{(h)} \right\|^2}{\left\| \mathbf{o}_q^{(h)} \right\|^2}, \tag{6}$$

where the normalization by $\|\mathbf{o}_q^{(h)}\|^2$ makes the contributions of different heads comparable. We read the attention weights from the self-attention of decoder layer $\ell$ using the last query position, so that the selection is conditioned on the instruction. Unlike importance-based criteria, Equation 6 is defined over subsets rather than individual tokens, so the value of a token depends on which tokens have already been selected.

Equation 6 is a combinatorial problem, and we approximate its solution greedily, which admits an efficient incremental update. Maintaining the running numerator and denominator of Equation 3 after $t$ selections,

$$\mathbf{n}_t = \mathbf{c}_q + \sum_{j \in \mathcal{S}_t} a_{q,j}\, \mathbf{v}_j, \qquad d_t = \tau_q + \sum_{j \in \mathcal{S}_t} a_{q,j}, \tag{7}$$

the attention output obtained by adding a candidate $j \notin \mathcal{S}_t$ is $(\mathbf{n}_t + a_{q,j}\mathbf{v}_j)/(d_t + a_{q,j})$, which requires only one vector addition and one scalar addition per candidate. All candidates are thus evaluated in a single broadcast operation, and $\mathbf{n}_t$ and $d_t$ are updated in place once a token is appended. We initialize $\mathcal{S}_1$ with $\arg\max_j a_{q,j}$, the token carrying the largest attention weight. This is the top-1 solution of attention-based pruning, so CentroidPrune starts from the attention-based selection and refines it by correcting the direction factor.

The procedure is sequential in $k$. We therefore fix the number of rounds to $T$ and append $\lceil k/T \rceil$ tokens per round, so that the number of iterations is independent of the budget. The objective and the initialization are unchanged; the only approximation is that tokens selected within the same round do not observe one another. With $T$ fixed, the resulting cost is $\mathcal{O}(N)$, linear in the number of visual tokens and independent of the budget, and accounts for less than $0.1\%$ of the FLOPs of a single decoder layer. Moreover, no additional forward pass is required, since $\mathbf{o}_q$, $a_{q,j}$ and $\mathbf{v}_j$ are all available from the attention computation of layer $\ell$. We use $T = 8$, and the complete procedure is summarized in Algorithm 1.

Rather than pruning at a single layer, we follow Xing et al. (2024) and partition the forward pass into two stages, applying Equation 6 at the end of each stage with a predefined retention ratio. The

Table 1: Performance Comparison on LLaVA-1.5-7B.

| Method | GQA Acc. ↑ | MMB Acc. ↑ | MME P+C ↑ | POPE F1 ↑ | SQA Acc. ↑ | VQA$^{V2}$ Acc. ↑ | VQA$^{Text}$ Acc. ↑ | Avg. ↑ |
|---|---|---|---|---|---|---|---|---|
| *Total 576 Tokens* | | | | | | | | |
| LLaVA-1.5-7B | 61.90 | 64.70 | 1862.00 | 85.90 | 69.50 | 78.50 | 58.20 | 100% |
| *Retain 192 Tokens ↓ 66.7%* | | | | | | | | |
| ToMe (ICLR 2023) | 54.30 | 60.50 | 1563.00 | 72.40 | 65.20 | 68.00 | 52.10 | 88.49% |
| FastV (ECCV 2024) | 52.70 | 61.20 | 1612.00 | 64.80 | 67.30 | 67.10 | 52.50 | 87.75% |
| HiRED (AAAI 2025) | 58.70 | 62.80 | 1737.00 | 82.80 | 68.40 | 74.90 | 47.40 | 93.84% |
| VisionZip (CVPR 2025) | 59.30 | 63.00 | 1782.60 | 85.30 | 68.90 | 76.80 | 57.30 | 97.66% |
| DivPrune (CVPR 2025) | 59.97 | 62.54 | 1762.23 | 87.00 | 68.91 | 76.87 | 56.97 | 97.78% |
| PyramidDrop (CVPR 2025) | 57.10 | 63.20 | 1766.00 | 82.30 | 68.80 | 75.10 | 56.10 | 95.95% |
| SparseVLM (ICML 2025) | 57.60 | 62.50 | 1721.00 | 83.60 | 69.10 | 75.60 | 56.10 | 95.93% |
| ZOO-Prune (CVPR 2026) | 59.91 | 63.32 | 1789.00 | **87.81** | 68.57 | **77.34** | 57.25 | 98.36% |
| **CentroidPrune (Ours)** | **60.88** | **63.40** | **1841.10** | 86.12 | **69.46** | 76.87 | **57.30** | **98.83%** |
| *Retain 128 Tokens ↓ 77.8%* | | | | | | | | |
| ToMe (ICLR 2023) | 52.40 | 53.30 | 1343.00 | 62.80 | 59.60 | 63.00 | 49.10 | 80.38% |
| FastV (ECCV 2024) | 49.60 | 56.10 | 1490.00 | 59.60 | 60.20 | 61.80 | 50.60 | 81.22% |
| HiRED (AAAI 2025) | 57.20 | 61.50 | 1710.00 | 79.80 | 68.10 | 73.40 | 46.10 | 91.84% |
| VisionZip (CVPR 2025) | 57.60 | 62.00 | 1761.70 | 83.20 | 68.90 | 75.60 | 56.80 | 96.20% |
| DivPrune (CVPR 2025) | 59.25 | 62.03 | 1718.22 | 86.72 | 68.96 | 75.96 | 56.06 | 96.73% |
| PyramidDrop (CVPR 2025) | 56.00 | 61.00 | 1644.00 | 82.30 | 68.30 | 72.90 | 55.10 | 93.52% |
| SparseVLM (ICML 2025) | 56.00 | 60.00 | 1696.00 | 80.50 | 67.10 | 73.80 | 54.90 | 93.27% |
| ZOO-Prune (CVPR 2026) | 59.36 | 62.11 | 1730.20 | **87.52** | 68.62 | **76.57** | 56.68 | 97.19% |
| **CentroidPrune (Ours)** | **60.84** | **63.32** | **1838.80** | 86.12 | **69.41** | 75.84 | **57.02** | **98.52%** |
| *Retain 64 Tokens ↓ 88.9%* | | | | | | | | |
| ToMe (ICLR 2023) | 48.60 | 43.70 | 1138.00 | 52.50 | 50.00 | 57.10 | 45.30 | 70.12% |
| FastV (ECCV 2024) | 46.10 | 48.00 | 1256.00 | 48.00 | 51.10 | 55.00 | 47.80 | 71.10% |
| HiRED (AAAI 2025) | 54.60 | 60.20 | 1599.00 | 73.60 | 68.20 | 69.70 | 44.20 | 87.95% |
| VisionZip (CVPR 2025) | 55.10 | 60.10 | 1690.00 | 77.00 | 69.00 | 72.40 | 55.50 | 92.74% |
| DivPrune (CVPR 2025) | 57.78 | 59.28 | 1674.40 | 85.56 | 68.17 | 74.11 | 54.69 | 94.42% |
| PyramidDrop (CVPR 2025) | 41.90 | 33.30 | 1092.00 | 55.90 | 68.60 | 69.20 | 45.90 | 72.66% |
| SparseVLM (ICML 2025) | 52.70 | 56.20 | 1505.00 | 75.10 | 62.20 | 68.20 | 51.80 | 86.52% |
| ZOO-Prune (CVPR 2026) | 58.47 | 59.62 | 1667.40 | **86.81** | 68.27 | 75.02 | 55.55 | 95.21% |
| **CentroidPrune (Ours)** | **60.78** | **63.32** | **1837.40** | 86.13 | **69.36** | **75.73** | **56.84** | **98.42%** |

sequence is truncated in place inside the decoder loop, so pruning is performed within a single forward pass. Implementation details are provided in Appendix A.2.

## 4 Experiments

In this section, we validate CentroidPrune across multiple vision-language architectures on comprehensive multimodal benchmarks.

**Models.** We evaluate the effectiveness of our method on three widely used open-source VLMs: LLaVA-1.5-7B (Liu et al., 2024a), LLaVA-NeXT-7B (Liu et al., 2024b), and Qwen2.5-VL-7B (Bai et al., 2025). LLaVA-1.5 represents a single image with 576 visual tokens, whereas LLaVA-NeXT extends it to higher resolution and produces up to 2,880 tokens per image. Qwen2.5-VL adopts a different architecture with dynamic resolution.

**Datasets.** We conduct experiments on diverse benchmarks: GQA (Hudson & Manning, 2019), MMBench (Liu et al., 2024c), MME (Fu et al., 2026), POPE (Li et al., 2023c), ScienceQA-IMG (Lu et al., 2022), VQAv2 (Goyal et al., 2017), TextVQA (Singh et al., 2019), MMMU (Yue et al., 2024), and SEED-Bench (Li et al., 2023a). These benchmarks cover compositional reasoning, perception, hallucination, scientific and knowledge-intensive reasoning, and text recognition, and thus probe complementary aspects of visual understanding.

Table 2: Performance Comparison on Qwen2.5-VL-7B.

| Method | GQA | MMB | MME | POPE | Avg. |
|---|---|---|---|---|---|
| *Baseline (Full Tokens)* | | | | | |
| Qwen2.5-VL-7B | 60.84 | 84.10 | 2310.00 | 86.30 | 100% |
| *Retain 20% Tokens* | | | | | |
| VisionZip (CVPR 2025) | 57.27 | 79.72 | 2221.00 | 83.89 | 95.6% |
| DivPrune (CVPR 2025) | **60.05** | 79.55 | 2173.00 | 83.42 | 96.0% |
| ZOO-Prune (CVPR 2026) | 58.81 | 80.60 | 2201.00 | 84.17 | 96.3% |
| **CentroidPrune (Ours)** | 59.57 | **82.04** | **2322.6** | **86.50** | **99.1%** |
| *Retain 10% Tokens* | | | | | |
| VisionZip (CVPR 2025) | 54.09 | 76.03 | 1937.00 | 78.97 | 88.7% |
| DivPrune (CVPR 2025) | 55.49 | 76.03 | 2054.00 | 79.05 | 90.5% |
| ZOO-Prune (CVPR 2026) | 55.45 | 76.28 | 2018.00 | 80.99 | 90.8% |
| **CentroidPrune (Ours)** | **59.45** | **81.79** | **2318.4** | **86.49** | **98.9%** |

### 4.1 Main Results

Across all three backbones, CentroidPrune shows substantially smaller performance degradation than existing methods as more visual tokens are removed.

**Results on LLaVA-1.5-7B.** As shown in Table 1, CentroidPrune achieves the best average performance at every budget, retaining 98.83%, 98.52%, and 98.42% of the unpruned model with 192, 128, and 64 visual tokens. Reducing the budget by a factor of three costs only 0.41 points, while ZOO-Prune and DivPrune lose 3.15 and 3.36 points over the same range. As a result, CentroidPrune with 64 tokens already surpasses every baseline evaluated with 192 tokens, and the margin over the second-best method reaches 3.21 points at 64 tokens.

**Results on LLaVA-NeXT-7B.** Table 3 shows a consistent trend under higher resolution and more aggressive pruning ratios. CentroidPrune retains 99.7%, 98.9%, and 98.4% of the unpruned model with 640, 320, and 160 tokens, improving over ZOO-Prune by 1.3, 1.9, and 3.1 points. At 160 tokens, which discards 94.4% of the visual tokens, it matches the average that ZOO-Prune attains with four times the budget.

**Results on Qwen2.5-VL-7B.** Table 2 evaluates a different architecture with dynamic resolution. Halving the retention ratio from 20% to 10% costs CentroidPrune only 0.2 points, whereas VisionZip and DivPrune lose 6.9 and 5.5 points. CentroidPrune retains 98.9% of the unpruned model at a 10% retention ratio, where all baselines fall below 91%.

### 4.2 Fidelity to the Unpruned Model

We analyze how closely each method reproduces the behavior of the unpruned model. We measure fidelity as $\mathrm{KL}(\text{unpruned} \,\|\, \text{pruned})$ over the next-token distribution at the answer position, computed over the full evaluation sets of GQA and TextVQA. Table 4 shows that CentroidPrune is the most faithful method on both benchmarks. On GQA, its KL divergence is 0.0780, against 0.1174 for ZOO-Prune and 0.1315 for DivPrune, and the margin is larger on TextVQA, where it reaches 0.0421 against 0.1830 and 0.3038. FastV, which selects tokens by attention magnitude alone, is the least faithful on both, with a KL divergence more than four times ours on GQA and nearly ten times ours on TextVQA. These results indicate that minimizing the change in the attention output indeed keeps the pruned model closer to the original one.

Table 3: Performance Comparison on LLaVA-NeXT-7B.

| Method | GQA Acc. ↑ | MMB Acc. ↑ | MME P+C ↑ | POPE F1 ↑ | SQA Acc. ↑ | VQA$^{V2}$ Acc. ↑ | VQA$^{Text}$ Acc. ↑ | MMMU Acc. ↑ | SEED-I Acc. ↑ | Avg. ↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| *Total 2880 Tokens* | | | | | | | | | | |
| LLaVA-NeXT-7B | 64.20 | 67.90 | 1842.00 | 86.40 | 70.20 | 80.10 | 61.30 | 35.10 | 70.20 | 100% |
| *Retain 640 Tokens ↓ 77.8%* | | | | | | | | | | |
| SparseVLM (ICML 2025) | 60.30 | 65.70 | 1772.00 | – | 67.70 | 77.10 | 57.80 | 34.60 | – | – |
| VisionZip (CVPR 2025) | 61.30 | 66.30 | 1787.00 | 86.30 | 68.10 | 79.10 | **60.20** | 34.70 | 66.70 | 97.5% |
| DivPrune (CVPR 2025) | 61.58 | 65.38 | 1773.04 | 85.51 | 67.82 | 78.94 | 55.41 | 36.89 | 67.56 | 97.1% |
| ZOO-Prune (CVPR 2026) | 61.94 | 65.21 | **1816.45** | 87.87 | 67.53 | **79.64** | 58.28 | 37.00 | 67.89 | 98.4% |
| **CentroidPrune (Ours)** | **63.69** | **66.58** | 1794.90 | **88.07** | **69.06** | 79.30 | 59.52 | **37.44** | **69.70** | **99.7%** |
| *Retain 320 Tokens ↓ 88.9%* | | | | | | | | | | |
| SparseVLM (ICML 2025) | 57.70 | 64.30 | 1694.00 | – | 67.30 | 73.40 | 55.90 | 34.40 | – | – |
| VisionZip (CVPR 2025) | 59.30 | 63.10 | 1702.00 | 82.10 | 67.30 | 76.20 | 58.90 | 35.30 | 63.40 | 94.5% |
| DivPrune (CVPR 2025) | 59.63 | 63.66 | 1731.04 | 83.47 | 67.82 | 76.64 | 53.84 | **37.11** | 65.35 | 95.1% |
| ZOO-Prune (CVPR 2026) | 61.20 | 64.43 | 1786.80 | 86.86 | 67.48 | 78.08 | 57.12 | 36.56 | 66.32 | 97.0% |
| **CentroidPrune (Ours)** | **63.56** | **66.58** | **1795.70** | **87.99** | **69.16** | **79.18** | **59.07** | 37.00 | **66.47** | **98.9%** |
| *Retain 160 Tokens ↓ 94.4%* | | | | | | | | | | |
| SparseVLM (ICML 2025) | 51.20 | 63.10 | 1542.00 | – | 67.50 | 66.30 | 46.40 | 32.80 | – | – |
| VisionZip (CVPR 2025) | 55.50 | 60.10 | 1630.00 | 74.80 | 68.30 | 71.40 | 56.20 | 36.10 | 58.30 | 90.4% |
| DivPrune (CVPR 2025) | 57.79 | 62.29 | 1658.25 | 79.36 | 68.02 | 73.92 | 52.42 | 36.44 | 62.54 | 92.4% |
| ZOO-Prune (CVPR 2026) | 59.77 | 63.75 | 1724.50 | 85.17 | 67.67 | 76.12 | 55.86 | 36.67 | 64.03 | 95.3% |
| **CentroidPrune (Ours)** | **63.54** | **66.75** | **1792.80** | **87.90** | **68.91** | **79.04** | **58.76** | **37.11** | **64.05** | **98.4%** |

Table 4: Fidelity to the unpruned model on LLaVA-1.5-7B when retaining 64 visual tokens. Fidelity is measured as KL(unpruned ‖ pruned) over the next-token distribution at the answer position.

| | GQA | | TextVQA | |
|---|---|---|---|---|
| **Method** | KL ↓ | Acc. ↑ | KL ↓ | Acc. ↑ |
| FastV (ECCV 2024) | 0.3638 | 46.10 | 0.4078 | 47.80 |
| DivPrune (CVPR 2025) | 0.1315 | 57.78 | 0.3038 | 54.69 |
| ZOO-Prune (CVPR 2026) | 0.1174 | 58.47 | 0.1830 | 55.55 |
| **CentroidPrune (Ours)** | **0.0780** | **60.78** | **0.0421** | **56.84** |

### 4.3 Efficiency Comparison

We evaluate efficiency on a single NVIDIA H200 GPU and report peak memory and end-to-end latency relative to the unpruned model in Figure 3. CentroidPrune attains the best accuracy–cost trade-off across both backbones and all budgets. It reduces peak memory to $0.80\times$ and end-to-end latency to $0.82\times$ while retaining $98.4\%$ of the unpruned performance. ZOO-Prune, which prunes before the LLM and would therefore be expected to be cheaper, instead requires $0.95\times$ peak memory because its zeroth-order estimation expands the visual features along $m$ perturbations at the projector, and recovers only $95.3\%$ at a comparable latency. CentroidPrune obtains its target from the attention computation of the pruning layer and thus introduces no such overhead. Fixing the number of rounds also keeps the selection cost independent of the budget, so its three budgets differ by less than $0.5\%$ in latency.

### 4.4 Ablation Study

We ablate the number of rounds $T$, which changes only how many tokens are appended per round while the objective and the initialization remain identical. As shown in Table 5, performance is stable across the range, varying by $0.70$ points between $T = 2$ and $T = 16$. Increasing $T$ improves the average, since fewer tokens are appended without observing the selections made alongside them, but the gain saturates and $T = 8$ is already within $0.30$ points of $T = 16$ while requiring half as many iterations. We therefore use $T = 8$.

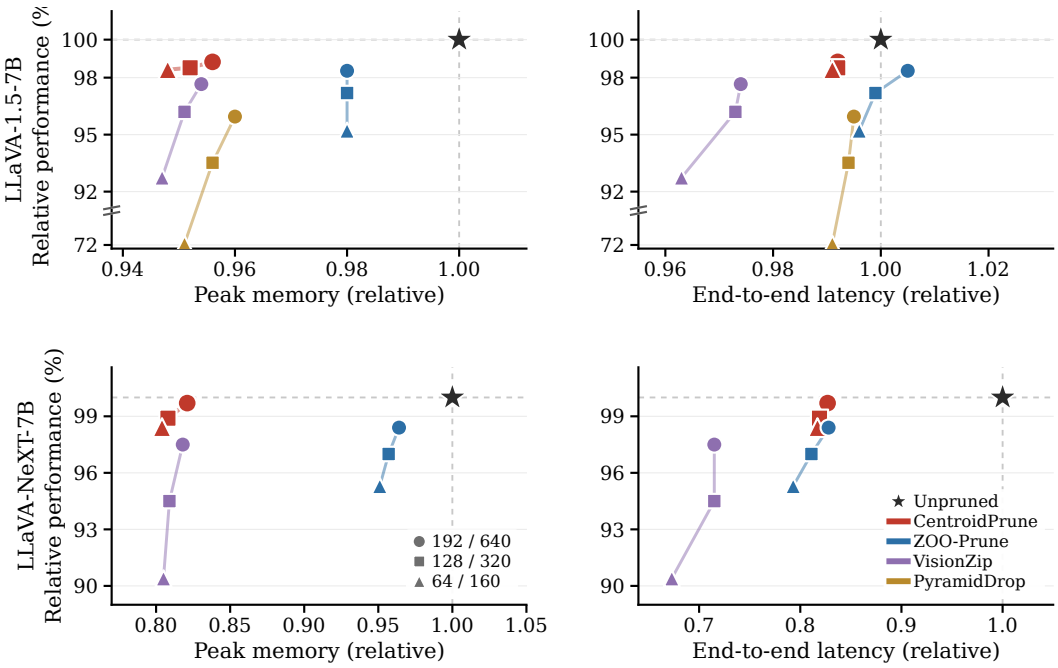


Figure 3: Performance against cost. The top row is LLaVA-1.5-7B and the bottom row is LLaVA-NeXT-7B; the left column is peak memory and the right column is end-to-end latency. Both are measured on a single NVIDIA H200 GPU and normalized by the unpruned model, marked with a star. Colour denotes the method and marker shape denotes the retained token budget. The vertical axis is the average performance normalized by the unpruned model.

Table 5: Effect of the number of rounds $T$ on LLaVA-1.5-7B, retaining 64 visual tokens.

| $T$ | **GQA** Acc. ↑ | **MMB** Acc. ↑ | **MME** P+C ↑ | **POPE** F1 ↑ | **SQA** Acc. ↑ | **VQA$^{V2}$** Acc. ↑ | **VQA$^{Text}$** Acc. ↑ | **Avg.** ↑ |
|---|---|---|---|---|---|---|---|---|
| *Total 576 Tokens* | | | | | | | | |
| – | 61.90 | 64.70 | 1862.00 | 85.90 | 69.50 | 78.50 | 58.20 | 100% |
| *Retain 64 Tokens* | | | | | | | | |
| 2 | 60.47 | 63.60 | 1838.2 | 84.39 | 68.77 | 75.24 | **57.25** | 98.02% |
| 4 | 60.64 | 63.00 | **1868.4** | 84.77 | 69.26 | 75.60 | 57.22 | 98.38% |
| 8 | 60.78 | 63.32 | 1837.4 | **86.13** | **69.36** | 75.73 | 56.84 | 98.42% |
| 16 | **60.88** | **64.00** | 1858.6 | 85.00 | **69.36** | **76.57** | 56.84 | **98.72%** |

## 5 Conclusion

In this paper, we revisit visual token pruning from the effect that pruning has on the attention output. We show that removing a set of visual tokens shifts the attention output by an amount that decomposes into two factors, the removed attention mass and the directional deviation of the removed tokens, and that attention-based methods minimize only the former. Based on this analysis, we propose CentroidPrune, a training-free pruning method that formulates token selection as a subset selection problem minimizing the distance between the original attention output and the output after pruning and renormalization. Extensive experiments across multiple VLM backbones and multimodal benchmarks demonstrate that CentroidPrune consistently outperforms prior methods, particularly under aggressive pruning ratios. We further show that it reproduces the output distribution of the unpruned model more closely than existing methods.

## AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing of this manuscript. We have not used generative AI tools for the design or implementation of the proposed method, or for the execution of the experiments. We have reviewed all AI-assisted work: every AI-assisted passage was read and revised by the authors, and all reported numbers were produced by our own experiments. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401. IEEE, 2025.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1773–1781, 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

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*, pp. 19–35. Springer, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 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*, pp. 24185–24198, 2024b.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Advances in Neural Information Processing Systems*, 38, 2026.

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. 2017.

Zonghao Guo, Ruyi Xu, Yuan Yao, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. In *European Conference on Computer Vision*, pp. 390–406. Springer, 2024.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702. IEEE, 2019.

Youngeun Kim, Youjia Zhang, Huiling Liu, Aecheon Jung, Sunwoo Lee, and Sungeun Hong. Zoo-prune: Training-free token pruning via zeroth-order gradient estimation in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39572–39582, 2026.

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

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

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023c.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26753–26763. IEEE, 2024.

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

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024b.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024c.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

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*, pp. 8748–8763. PmLR, 2021.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22857–22867. IEEE, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8309–8318. IEEE, 2019.

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.

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.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802. IEEE, 2025.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9556–9567, 2024.

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*, pp. 881–916, 2025.

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

# A Appendix

## A.1 Datasets

**GQA** (Hudson & Manning, 2019) evaluates compositional visual reasoning with questions generated from the scene graphs of real-world images, requiring the model to reason over objects, attributes, and their spatial and relational structure. We evaluate on the balanced test-dev split.

**MMBench** (Liu et al., 2024c) is a multiple-choice benchmark that covers a broad range of perception and reasoning abilities across fine-grained dimensions. We evaluate on the English development split.

**MME** (Fu et al., 2026) measures perception and cognition through yes/no questions spanning subtasks such as existence, counting, position, color, OCR, and commonsense or numerical reasoning. We report the sum of the perception and cognition scores.

**POPE** (Li et al., 2023c) probes object hallucination by asking whether a given object is present in the image, with negative objects sampled under random, popular, and adversarial settings. We report the F1 score.

**ScienceQA-IMG** (Lu et al., 2022) consists of multiple-choice science questions covering natural, social, and language science. We use the subset whose questions are accompanied by an image.

**VQAv2** (Goyal et al., 2017) is an open-ended visual question answering benchmark in which each question is paired with two images yielding different answers, so that language priors alone are insufficient.

**TextVQA** (Singh et al., 2019) requires reading and reasoning over text appearing in the image, and therefore depends on visual evidence concentrated in small regions.

**MMMU** (Yue et al., 2024) contains college-level questions across multiple disciplines that require domain knowledge together with the interpretation of figures, charts, and diagrams.

**SEED-Bench** (Li et al., 2023a) is a multiple-choice benchmark spanning several evaluation dimensions of spatial and temporal understanding. We use the image subset.

## A.2 Implementation Details

**Pruning schedule.** We apply Equation 6 at layers $8$ and $16$ of the decoder. The first stage retains $50\%$ of the visual tokens, and the second stage retains the target budget. Since pruning is performed during prefill, the key-value cache of the subsequent layers is built at the reduced length and reused throughout decoding. The same schedule is used for all backbones and all benchmarks, and the number of rounds is fixed to $T = 8$ at both stages. For backbones whose number of visual tokens varies with the input image, the budget is specified as a retention ratio rather than an absolute count, so that the same fraction is retained regardless of the input resolution.

**Obtaining the attention signal.** At each pruning stage, we read the self-attention of the corresponding decoder layer at the last query position. Attention is requested only for that single layer,

so the remaining layers run with the default attention implementation. The value vectors $\mathbf{v}_j$ are the ones computed inside the same layer, that is, after its layer normalization and value projection, rather than the projector output. Since the layer normalization preceding the value projection is not linear, the two do not lead to the same selection.

**Evaluation setup.** All accuracy numbers are obtained with lmms-eval (Zhang et al., 2025) under a single codebase. Efficiency is measured on a single NVIDIA H200 GPU. End-to-end latency is reported for 32 generated tokens, and peak memory is the maximum allocated device memory over the forward pass. CentroidPrune is training-free and requires no calibration data.