# What You Say is What You See: Anchoring Visual Token Pruning on Textual Essentials for Efficient LVLM Inference

Anonymous ACL submission

## Abstract

Processing lengthy sequences of visual tokens incurs substantial computational overhead, presenting a critical bottleneck for Large Vision-Language Models (LVLMs). Existing token pruning methods face a fundamental dilemma: text-agnostic approaches ignore user instructions, while attention-based techniques suffer from *text-visual semantic misalignment*, where cross-attention maps fail to reliably localize query-relevant regions. To overcome these limitations, we introduce **TextScythe**, a novel plug-and-play framework that reframes compression as *instruction distillation*. Our core insight is to first distill the user instruction into a minimal set of *vision-critical text tokens* using a novel Entropy-Ratio (ER) metric, which quantifies the specificity and salience of cross-modal semantic correspondence. These distilled tokens then serve as precise anchors to select semantically relevant visual patches, after which a diversity-preserving mechanism supplements representative background tokens to maintain global context. This "understand-then-prune" paradigm ensures accurate alignment with user intent while effectively suppressing visual noise. Extensive experiments on 12 image and video benchmarks demonstrate that TextScythe achieves highly efficient compression, retaining **96.6%** of the original accuracy while pruning up to **88.9%** of visual tokens for LLaVA-1.5. The framework shows robust generalization across diverse VLM architectures and high-resolution settings, offering a practical acceleration solution without any modification to the transformer internals.

## 1 Introduction

Large Vision-Language Models (LVLMs) have demonstrated remarkable capabilities across a diverse array of multimodal tasks, from visual question answering and reasoning to detailed image description (Liu et al., 2024c; Wang et al., 2024b). This success, however, is predicated on processing lengthy sequences of visual tokens, particularly for high-resolution images (Li et al., 2024c) and long videos (Lin et al., 2023), which incurs substantial computational and memory costs. These demands limit the practical deployment of LVLMs in resource-constrained environments (Liu et al., 2024a). A key observation mitigating this bottleneck is that for a given user instruction, a significant portion of visual tokens are either semantically redundant or irrelevant, presenting a prime opportunity for compression through token pruning.

Existing efforts to reduce LVLM inference cost via visual token pruning can be broadly categorized into two paradigms, each with fundamental limitations. The first paradigm operates in a *text-agnostic* manner, identifying and merging redundant tokens based on intra-modal visual feature similarity (Wen et al., 2025b; Alvar et al., 2025) or [CLS] attention (Zou et al., 2025a). While effective at reducing spatial redundancy, such methods risk discarding tokens that are critical to the specific user query. The second paradigm is *query-aware*, leveraging cross-attention weights within the Language Model (LLM) to identify important tokens (Chen et al., 2024; Zhang et al., 2024c). However, we identify that these attention-based methods suffer from a critical issue of *text-visual semantic misalignment*: the cross-attention mechanism often produces diffuse or biased attention maps that fail to accurately pinpoint image regions semantically relevant to the query. As illustrated in Figure 1, this misalignment can lead to misguided pruning, retaining irrelevant patches while discarding crucial ones.

The core challenge, therefore, is to establish a *reliable and instruction-aware signal* for assessing visual token importance. Our work is driven by two pivotal insights. First, we demonstrate that the cosine similarity between text and visual embeddings provides a more geometrically faithful measure of semantic relevance than attention maps, which are often confounded by positional biases


Figure 1: Case studies of our proposed TextScythe *vs.* attention-based pruning: TextScythe resolves the problem of text-visual misalignment in previous pruning strategies, and accurately preserves query-relevant visual tokens.

(Sec. 2). Second, we recognize that not all words in an instruction are equally useful for guiding visual selection; common words introduce noise. This motivates a paradigm shift: instead of pruning directly with a noisy or misaligned global signal, we propose to first *distill* the user's instruction into a minimal set of robust semantic anchors, and then use these distilled anchors to guide precise pruning.

To this end, we introduce **TextScythe**, a plug-and-play visual token pruning framework that embodies an *instruction distillation and guided pruning* paradigm. TextScythe operates in two stages: (1) it employs a novel *Entropy-Ratio (ER)* metric to dynamically distill the user instruction into a compact set of *vision-critical text tokens*; (2) these key tokens then guide the selection of semantically relevant visual tokens, after which a diversity-aware mechanism supplements representative background tokens to preserve global context. This "***understand-then-prune***" workflow ensures precise alignment with user intent. As illustrated in Figure 2, TextScythe maintains robust performance even under high pruning ratios, significantly outperforming existing methods.

In summary, our contributions are threefold:

- We identify and quantitatively analyze the *text-visual semantic misalignment* problem and advocate for semantic cosine similarity as a more robust signal for cross-modal relevance.
- We introduce a novel *instruction distillation* paradigm centered on the ***ER*** metric, which dynamically extracts vision-critical text tokens to serve as precise anchors for token pruning.
- We instantiate this paradigm in the **TextScythe** framework, which integrates instruction-aware token selection with diversity-aware background supplementation. Extensive experiments show that our method achieves state-of-the-art results, excelling in both performance and efficiency across diverse benchmarks and models.

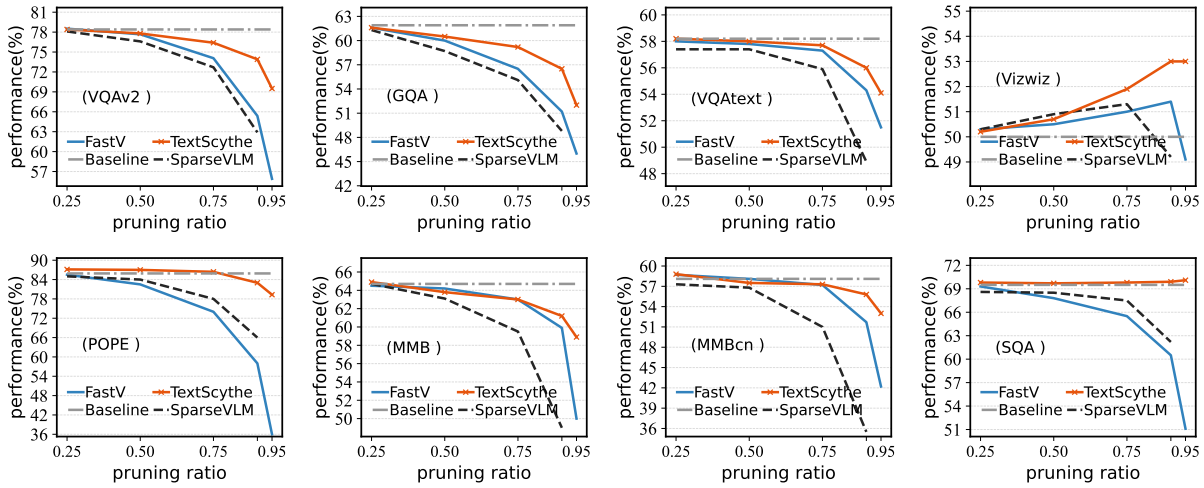


Figure 2: Performance of different baselines under varying pruning ratios. As the token pruning ratio grows, the performance of these attention-based strategies degrades dramatically, while TextScythe maintains the best results.

## 2 Motivation: From Signal Misalignment to Instruction Distillation

**The Pitfall of Existing Pruning Signals.** Accurately assessing the importance of a visual token with respect to a user instruction is the cornerstone of effective pruning. However, current methodologies rely on signals that are fundamentally flawed. As mentioned above, intra-modal similarity or [CLS] attention is instruction-agnostic. Conversely, while cross-attention weights are instruction-aware, our analysis reveals they are an unreliable guide due to *text-visual semantic misalignment*. This misalignment stems from the attention mechanism's primary objective: to holistically blend multimodal features for fusion, not to perform precise, spatially-grounded localization. Consequently, attention maps are often fail to concentrate on semantically pertinent regions, as visually corroborated in Fig. 3.

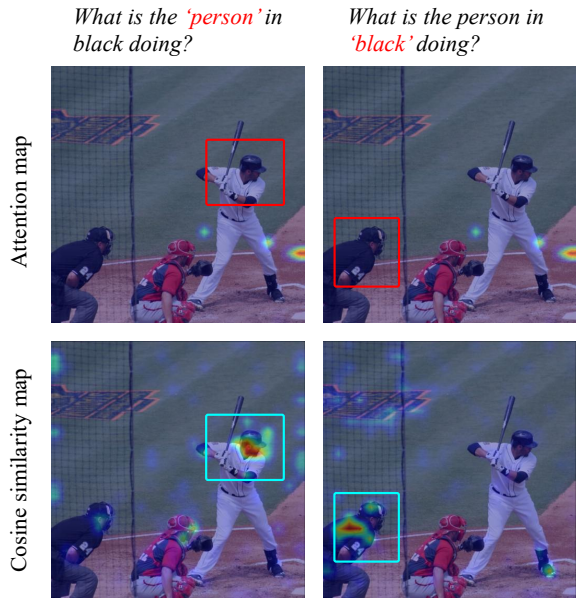


Figure 3: Comparison of attention and cosine similarity visualizations between key text and image.

We further quantify this core limitation through a large-scale analysis on the MME benchmark (Fu et al., 2023). As shown in Fig. 4, averaging cross-attention scores for each visual token position across diverse samples reveals a pronounced **systematic positional bias**: a small, fixed set of token indices consistently receives disproportionately high attention, irrespective of the actual image content or user query. This pattern indicates that cross-attention is often confounded by low-level positional priors inherent in the transformer's architecture, acting as a noisy, location-sensitive heuristic rather than a faithful semantic relevance signal.

In stark contrast, the cosine similarity between text and visual embeddings exhibits a markedly different profile. Its distribution across token positions is significantly more uniform. This demonstrates that cosine similarity is more directly governed by semantic correspondence, effectively mitigating the positional bias that plagues attention maps.

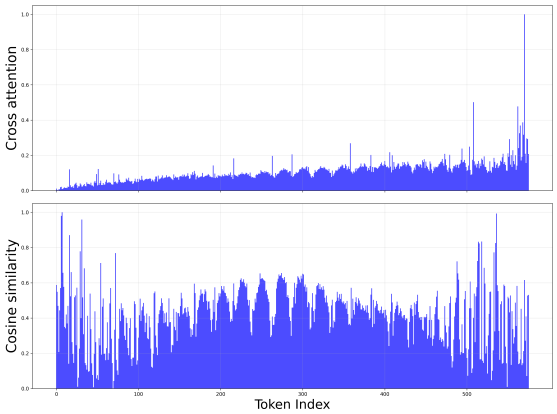


Figure 4: Comparison of average attention and cosine similarity between key text and image distributions across visual tokens on the MME dataset.

**Correction and Distillation.** To address the limitations of existing pruning signals, we propose a dual strategy. First, we employ cosine similarity between text and visual embeddings to correct the geometric misalignment inherent in cross-attention. However, computing similarity against all words in an instruction remains problematic, as not every text token contributes equally to localizing key image regions. As shown in Fig. 5 (Right), common words (*e.g.*, "there," "this") introduce noise and can adversely affect the selection of correct visual tokens. This leads to our core insight: effective pruning must first understand the instruction. We therefore distill the query into a minimal set of vision-critical text tokens and use only these distilled tokens to guide visual selection, a paradigm we term "***understand-then-prune***". Our ER metric operationalizes this distillation by identifying tokens with strong, specific visual correspondence while filtering out generic ones (Sec. 3.1).

## 3 Methodology

Building on the above analysis, we propose TextScythe, which retains visual tokens with strong semantic alignment to the instruction text. By pruning redundant visual tokens before the LLM decoder, TextScythe reduces computational cost while maintaining task-critical visual information. The framework overview is shown in Figure 7.

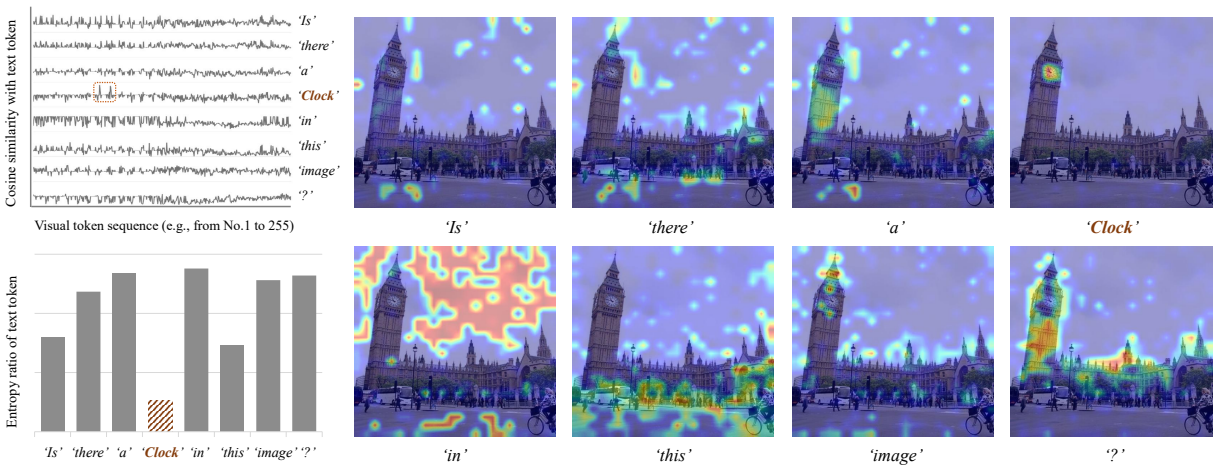


Figure 5: (Top-left) Line plots of cosine similarity for each text token against all visual tokens. (Bottom-left) The computed ER value for each text token. Those with anomalously low ER values are identified as key text tokens.

### 3.1 ER Metric: A Robust Distillation Filter

To identify which text tokens are semantically important for visual tokens, we propose the Entropy-Ratio (ER) metric. This metric quantifies how specifically a text token is associated with image. Given projected visual features $\mathbf{V} \in \mathbb{R}^{N_v \times D}$ and instruction text embeddings $\mathbf{T} \in \mathbb{R}^{N_t \times D}$, for each text token $i$, we obtain a normalized distribution over visual tokens:

$$\mathbf{P}_i = \text{softmax}(\mathbf{S}_i) = \text{softmax}(\mathbf{T}\mathbf{V}^\top). \quad (1)$$

where $\mathbf{S} \in \mathbb{R}^{N_t \times N_v}$ denotes cosine similarity matrix. ER metric for token $i$ is then defined as:

$$\text{ER}_i = \frac{E_i}{R_i} = \frac{-\sum_j \mathbf{P}_{ij} \log \mathbf{P}_{ij}}{\max_j(\mathbf{P}_{i:}) \,/\, \text{mean}(\mathbf{P}_{i:})}. \quad (2)$$

Here, The numerator $E_i$ is the entropy of the similarity distribution between the text token and visual tokens. A low $E_i$ indicates that the text token has a concentrated high similarity distribution, meaning it is strongly associated with specific regions in the image (Fig 5 Top-left). The denominator $R_i$ is the ratio between the maximum and average similarity. Since text tokens that match objects in the image tend to have a prominent similarity peak, while unmatched tokens have a more uniform similarity distribution, $R_i$ further distinguishes the ER values of image-relevant text tokens from others (Fig 5 Bottom-left).

We empirically validate ER on COCO 2014 (Lin et al., 2014). As shown in Fig. 6, when an image contains a target object (*e.g.*, an apple), the corresponding text token attains a minimal ER value; when the object is absent, the same token's ER increases significantly. This indicates that ER can reliably identify text tokens that correspond to specific objects in the image, confirming its effectiveness in detecting vision-critical tokens.

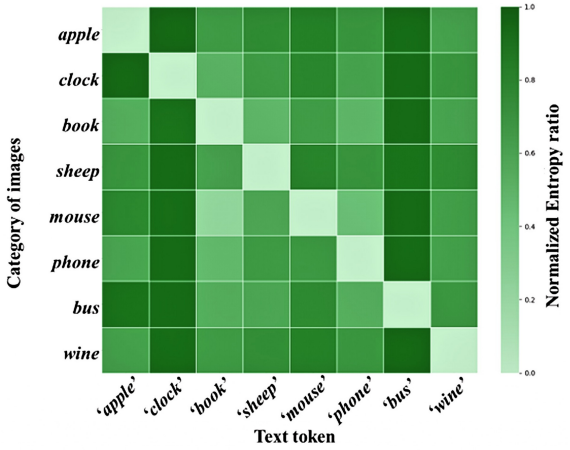


Figure 6: The entropy ratio between the object's text token and tokens of images containing different objects.

### 3.2 The TextScythe Framework

TextScythe first identifies vision critical text tokens using the ER metric, then selects visual tokens based on these critical text tokens. The process ensures that selected visual tokens are both relevant to the instruction and visually diverse.

#### 3.2.1 Vision critical Text Token Distillation

To automatically identify text tokens that anchor the visual semantics, we employ adaptive thresholding on the ER distribution. For a sequence of $N_t$ text tokens with ER values $\{\text{ER}_i\}_{i=1}^{N_t}$, we compute the mean $\mu_{\text{ER}}$ and standard deviation $\sigma_{\text{ER}}$. The threshold is set as:

$$\tau_{\text{ER}} = \mu_{\text{ER}} - \lambda \cdot \sigma_{\text{ER}}, \quad (3)$$

where $\lambda = \alpha \times (\lfloor \log_{10}(N_t) \rceil + 1)$, $\lfloor \cdot \rceil$ denotes rounding to the nearest integer and $\alpha$ being an em-

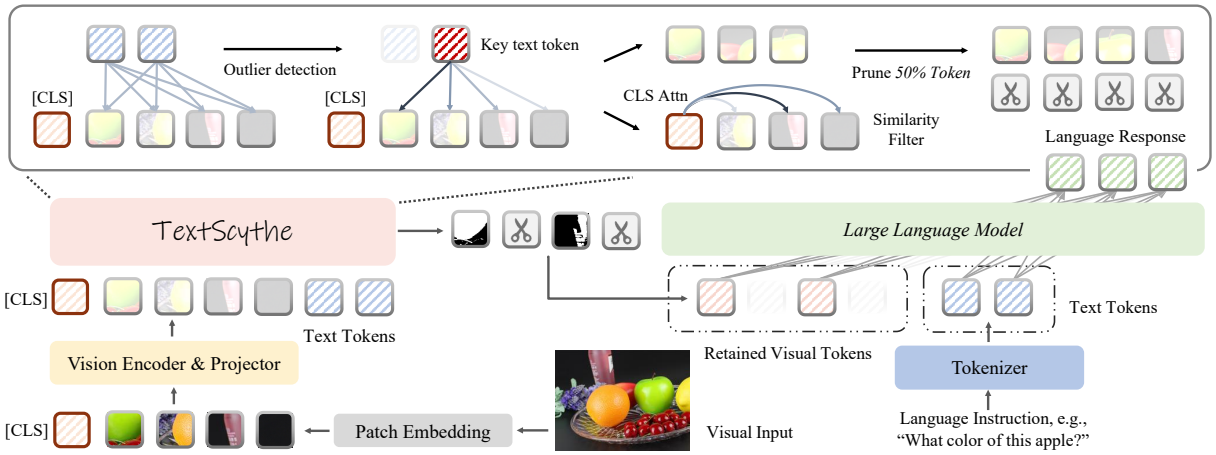


Figure 7: TextScythe first identifies key text tokens using the entropy of cross-modal cosine similarity. Then, it selects relevant visual tokens based on the similarity between key text tokens and visual tokens. Subsequently, TextScythe adds those with high attention visual tokens to enhance the completeness of the visual context.

pirically determined scaling factor. The choice of $\alpha$ is discussed in Appendix A.3. The adaptive coefficient $\lambda$ adjusts based on instruction length: longer instructions typically contain more non-visual text tokens, requiring a stricter threshold, while shorter instructions with fewer non-visual tokens allow a more lenient threshold. This ensures robust filtering across different query lengths. Text tokens with $\mathrm{ER}_i < \tau_{\mathrm{ER}}$ are selected as key semantic anchors:

$$\mathcal{T}_{\text{key}} = \{t_i \mid \mathrm{ER}_i < \tau_{\mathrm{ER}}\}. \tag{4}$$

These key tokens act as anchors to retrieve the most relevant visual tokens in the next stage.

#### 3.2.2 Query sensitive Visual Token Selection

After identifying the key text tokens $\mathcal{T}_{\text{key}}$, we proceed to select visual tokens that are strongly associated with them. For each key token $t_i \in \mathcal{T}_{\text{key}}$, we compute the mean $\mu_i$ and standard deviation $\sigma_i$ of its similarity distribution $\mathbf{P}_i$. Those probability exceeds an adaptive threshold tokens are selected:

$$\mathcal{V}^i_{\text{rel}} = \{v_j \mid \mathbf{P}_{ij} > \mu_i + \lambda \cdot \sigma_i\}. \tag{5}$$

We then take the union of all visual tokens that are strongly associated with the key text tokens:

$$\mathcal{V}_{\text{rel}} = \bigcup_{t_i \in \mathcal{T}_{\text{key}}} \mathcal{V}^i_{\text{rel}}. \tag{6}$$

#### 3.2.3 Diversity aware supplementation.

In addition to the instruction-relevant tokens, we supplement $\mathcal{V}_{\text{rel}}$ with background tokens from the remaining set $\mathcal{R} = \mathcal{V} \setminus \mathcal{V}_{\text{rel}}$ to prevent excessive information loss and enhance scene understanding.

We leverage the vision encoder's attention mechanism to assess token importance. For encoders with a [CLS] token, we use its attention weights over other tokens; otherwise, we compute mean attention across all tokens, denoted as $\mathbf{A}$.

To ensure both importance and diversity, we employ an iterative selection strategy. First, the token with the highest attention score in $\mathcal{R}$ is added to the supplementary set $\mathcal{V}_{\text{sup}}$. For subsequent selections, we compute a comprehensive score for each candidate $v_j \in \mathcal{R}$:

$$\text{score}(v_j) = \mathbf{A}_j - \max_{v_k \in \mathcal{V}_{\text{sup}}} \cos(\mathbf{V}_j, \mathbf{V}_k), \tag{7}$$

where $\cos(\mathbf{V}_j, \mathbf{V}_k)$ denotes the cosine similarity between visual tokens $j$ and $k$. The candidate maximizing this score is added to $\mathcal{V}_{\text{sup}}$; the process repeats until the total number of selected tokens reaches the budget $K$.

## 4 Experiments

**Experiment Setting.** We conduct experiments on multiple LVLMs across a range of benchmarks. Implementation details are provided in Appendix A.1.

### 4.1 Main Results

We evaluate TextScythe on LLaVA-1.5 across nine multimodal benchmarks under three aggressive pruning ratios (66.7%, 77.8%, and 88.9%). As shown in Table 1, TextScythe consistently outperforms all competing methods, including the recent HoloV, across every compression level. For instance, when removing 88.9% of visual tokens, TextScythe incurs only a 3.4% average performance drop, which is lower than the 4.2% drop of HoloV and the 6.1% drop of DART. At the moderate pruning ratios of 66.7% and 77.8%, the

Table 1: Performance comparison of various methods across different benchmarks. Results are shown for different pruning ratios, with accuracy and average performance highlighted. The best results in blue.

| Methods | GQA | MMB | $MMB_{CN}$ | MME | POPE | SQA | $VQA_{V2}$ | $VQA_{Text}$ | VizWiz | Average |
|---|---|---|---|---|---|---|---|---|---|---|
| Upper Bound, 576 Tokens | 61.9 | 64.7 | 58.1 | 1862 | 85.9 | 69.5 | 78.4 | 58.2 | 50.0 | 100% |
| LLaVA-1.5-7B | *Budget = 192 Tokens; Token Pruning Rate = 66.7%* | | | | | | | | | |
| ToMe (ICLR23) | 54.3 | 60.5 | - | 1563 | 72.4 | 65.2 | 68.0 | 52.1 | - | 88.5% |
| FastV (ECCV24) | 52.7 | 61.2 | 57.0 | 1612 | 64.8 | 67.3 | 67.1 | 52.5 | 50.8 | 90.5% |
| LLaVA-PruMerge (2024.5) | 54.3 | 59.6 | 52.9 | 1632 | 71.3 | 67.9 | 70.6 | 54.3 | 50.1 | 91.4% |
| PDrop (2024.10) | 57.1 | 63.2 | 56.8 | 1766 | 82.3 | 68.8 | 75.1 | 56.1 | 51.1 | 96.7% |
| FiCoCo-V (2024.11) | 58.5 | 62.3 | 55.3 | 1732 | 82.5 | 67.8 | 74.4 | 55.7 | 51.0 | 96.1% |
| MustDrop (2024.11) | 58.2 | 62.3 | 55.8 | 1787 | 82.6 | 69.2 | 76.0 | 56.5 | 51.4 | 97.2% |
| HiRED (AAAI25) | 58.7 | 62.8 | 54.7 | 1737 | 82.8 | 68.4 | 74.9 | 47.4 | 50.1 | 94.6% |
| SparseVLM (2025.2) | 57.6 | 62.5 | 53.7 | 1721 | 83.6 | 69.1 | 75.6 | 56.1 | 50.5 | 96.1% |
| DART (2025.2) | 58.9 | 63.6 | 57.0 | 1856 | 82.8 | 69.8 | 76.7 | 57.4 | 51.1 | 98.5% |
| HoloV (NeurIPS25) | 59.0 | 65.4 | 58.0 | 1820 | 85.6 | 69.8 | 76.7 | 57.4 | 50.9 | 99.2% |
| TextScythe (Ours) | 60.0 | 63.1 | 57.3 | 1798 | 87.2 | 69.8 | 77.3 | 57.8 | 51.6 | 99.2% |
| LLaVA-1.5-7B | *Budget = 128 Tokens; Token Pruning Rate = 77.8%* | | | | | | | | | |
| ToMe (ICLR23) | 52.4 | 53.3 | - | 1343 | 62.8 | 59.6 | 63.0 | 49.1 | - | 80.4% |
| FastV (ECCV24) | 49.6 | 56.1 | 56.4 | 1490 | 59.6 | 60.2 | 61.8 | 50.6 | 51.3 | 85.4% |
| LLaVA-PruMerge (2024.5) | 53.3 | 58.1 | 51.7 | 1554 | 67.2 | 67.1 | 68.8 | 54.3 | 50.3 | 89.4% |
| PDrop (2024.10) | 56.0 | 61.1 | 56.6 | 1644 | 82.3 | 68.3 | 72.9 | 55.1 | 51.0 | 94.9% |
| FiCoCo-V (2024.11) | 57.6 | 61.1 | 54.3 | 1711 | 82.2 | 68.3 | 73.1 | 55.6 | 49.4 | 94.9% |
| MustDrop (2024.11) | 56.9 | 61.1 | 55.2 | 1745 | 78.7 | 68.5 | 74.6 | 56.3 | 52.1 | 95.7% |
| HiRED (AAAI25) | 57.2 | 61.5 | 53.6 | 1710 | 79.8 | 68.1 | 73.4 | 46.1 | 51.3 | 93.1% |
| SparseVLM (2025.2) | 56.0 | 60.0 | 51.1 | 1696 | 80.5 | 67.1 | 73.8 | 54.9 | 51.4 | 93.8% |
| DART (2025.2) | 57.9 | 63.2 | 57.0 | 1845 | 80.1 | 69.1 | 75.9 | 56.4 | 51.7 | 97.5% |
| HoloV (NeurIPS25) | 57.7 | 63.9 | 56.5 | 1802 | 84.0 | 69.8 | 75.5 | 56.8 | 51.5 | 98.0% |
| TextScythe (Ours) | 59.1 | 62.5 | 56.9 | 1787 | 86.4 | 69.8 | 76.4 | 57.2 | 52.2 | 98.6% |
| LLaVA-1.5-7B | *Budget = 64 Tokens; Token Pruning Rate = 88.9%* | | | | | | | | | |
| ToMe (ICLR23) | 48.6 | 43.7 | - | 1138 | 52.5 | 50.0 | 57.1 | 45.3 | - | 70.1% |
| FastV (ECCV24) | 46.1 | 48.0 | 52.7 | 1256 | 48.0 | 51.1 | 55.0 | 47.8 | 50.8 | 76.7% |
| LLaVA-PruMerge (2024.5) | 51.9 | 55.3 | 49.1 | 1549 | 65.3 | 68.1 | 67.4 | 54.0 | 50.1 | 87.7% |
| PDrop (2024.10) | 41.9 | 33.3 | 50.5 | 1092 | 55.9 | 68.6 | 69.2 | 45.9 | 50.7 | 77.5% |
| FiCoCo-V (2024.11) | 52.4 | 60.3 | 53.0 | 1591 | 76.0 | 68.1 | 71.3 | 53.6 | 49.8 | 91.5% |
| MustDrop (2024.11) | 53.1 | 60.0 | 53.1 | 1612 | 68.0 | 63.4 | 69.3 | 54.2 | 51.2 | 90.1% |
| HiRED (AAAI25) | 54.6 | 60.2 | 51.4 | 1599 | 73.6 | 68.2 | 69.7 | 44.2 | 50.2 | 89.4% |
| SparseVLM (2025.2) | 52.7 | 56.2 | 46.1 | 1505 | 75.1 | 62.2 | 68.2 | 51.8 | 50.1 | 87.3% |
| DART (2025.2) | 55.9 | 60.6 | 53.2 | 1765 | 73.9 | 69.8 | 72.4 | 54.4 | 51.6 | 93.9% |
| HoloV (NeurIPS25) | 55.3 | 63.3 | 55.1 | 1715 | 80.3 | 69.5 | 72.8 | 55.4 | 52.8 | 95.8% |
| TextScythe (Ours) | 56.5 | 61.2 | 55.7 | 1727 | 83.0 | 69.9 | 73.9 | 56.0 | 53.4 | 96.6% |

performance drops are merely 0.8% and 1.4%, respectively, demonstrating robust retention of accuracy even under substantial token reduction. On detailed visual question answering benchmarks such as VizWiz and SQA, TextScythe not only preserves but occasionally exceeds the accuracy of the unpruned baseline, indicating that the removal of irrelevant visual noise can in fact sharpen the model's focus and enhance fine-grained understanding. Furthermore, TextScythe shows a pronounced advantage on hallucination-sensitive evaluation. On the POPE benchmark under 88.9% pruning, it achieves an accuracy of 83.0, outperforming the second-best method by nearly 3 points. This result highlights the method's capability to preserve semantically critical tokens that are essential for mitigating model hallucination, thus validating the effectiveness of the ER-based distillation strategy.

### 4.2 High-Resolution & Video Understanding

To assess scalability under high visual redundancy, we evaluate TextScythe on LLaVA-NeXT-7B, which processes high-resolution images, generating sequences of up to 2,880 visual tokens. Under a constrained budget of 320 tokens (88.9% pruned), TextScythe achieves an average performance retention of 95.8%, maintaining a competitive edge over the strong baseline HoloV (95.6%) and surpassing other methods including DART (93.9%) (Table 2). Notably, TextScythe obtains the highest scores on several key benchmarks including MME (1771), POPE (86.8), and SQA (71.6), demonstrating its effectiveness in preserving both fine-grained details and global semantics under extreme compression.

TextScythe also generalizes robustly to video understanding. As shown in Table 3, when applied to Video-LLaVA with 50% of tokens retained, it

Table 2: Performance comparison of various methods across different benchmarks. Results are shown for different pruning ratios, with accuracy and average performance highlighted. The best results in blue.

| Methods | GQA | MMB | $MMB_{CN}$ | MME | POPE | SQA | $VQA_{V2}$ | $VQA_{Text}$ | VizWiz | Average |
|---|---|---|---|---|---|---|---|---|---|---|
| Upper Bound, 2880 Tokens | 64.2 | 67.4 | 60.6 | 1851 | 86.5 | 70.1 | 81.8 | 64.9 | 57.6 | 100% |
| LLaVA-NeXT-7B | *Retain 320 Tokens* (↓ 88.9%) | | | | | | | | | |
| FastV (ECCV24) | 55.9 | 61.6 | 51.9 | 1661 | 71.7 | 62.8 | 71.9 | 55.7 | 53.1 | 88.0% |
| LLaVA-PruMerge (2024.5) | 53.6 | 61.3 | 55.3 | 1534 | 60.8 | 66.4 | 69.7 | 50.6 | 54.0 | 85.6% |
| PDrop (2024.10) | 56.4 | 63.4 | 56.2 | 1663 | 77.6 | 67.5 | 73.5 | 54.4 | 54.1 | 90.9% |
| MustDrop (2024.11) | 57.3 | 62.8 | 55.1 | 1641 | 82.1 | 68.0 | 73.7 | 59.9 | 54.0 | 92.2% |
| FasterVLM (ICCV25) | 56.9 | 61.6 | 53.5 | 1701 | 83.6 | 66.5 | 74.0 | 56.5 | 52.6 | 91.1% |
| HiRED (AAAI25) | 59.3 | 64.2 | 55.9 | 1690 | 83.3 | 66.7 | 75.7 | 58.8 | 54.2 | 93.3% |
| SparseVLM (2025.2) | 56.1 | 60.6 | 54.5 | 1533 | 82.4 | 66.1 | 71.5 | 58.4 | 52.0 | 89.7% |
| $GlobalCom^2$ (2025.3) | 57.1 | 61.8 | 53.4 | 1698 | 83.8 | 67.4 | 76.7 | 57.2 | 54.6 | 92.2% |
| DART (EMNLP25) | 61.7 | 65.3 | 58.2 | 1710 | 84.1 | 68.4 | 79.1 | 58.7 | 56.1 | 93.9% |
| HoloV (NeurIPS25) | 61.7 | 65.3 | 57.5 | 1738 | 83.9 | 68.9 | 79.5 | 58.7 | 55.3 | 95.6% |
| TextScythe (Ours) | 60.0 | 65.4 | 56.5 | 1771 | 86.8 | 71.6 | 77.2 | 59.2 | 54.8 | 95.8% |

performs competitively with the strongest baseline DART and outperforms other efficient methods like FastV. This confirms that our instruction-guided pruning paradigm remains effective for sequential visual data, handling both spatial and temporal redundancy without task-specific adaptation.

Table 3: Video QA Evaluations under 50% of visual tokens.

| Methods | TGIF-QA | | MSVD-QA | | MSRVT-QA | | Avgerge | |
|---|---|---|---|---|---|---|---|---|
| | Acc. | Score | Acc. | Score | Acc. | Score | Acc. | Score |
| LLaMA-Adapter | - | - | 54.9 | 3.1 | 43.8 | 2.7 | - | - |
| VideoChat | 34.4 | 2.3 | 56.3 | 2.8 | 45.0 | 2.5 | 45.1 | 2.5 |
| Video-LLaMA | - | - | 51.6 | 2.5 | 29.6 | 1.8 | - | - |
| Video-ChatGPT | 51.4 | 3.0 | 64.9 | 3.3 | 49.3 | 2.8 | 55.2 | 3.0 |
| Video-LLaVA | 47.0 | 3.4 | 70.2 | 3.9 | 57.3 | 3.5 | 58.2 | 3.6 |
| FastV (ECCV24) | 45.2 | 3.1 | 71.0 | 3.9 | 55.0 | 3.5 | 57.1 | 3.5 |
| DART (EMNLP25) | 46.3 | 3.3 | 71.0 | 4.0 | 56.7 | 3.6 | 58.0 | 3.7 |
| TextScythe (Ours) | 46.2 | 3.4 | 70.8 | 3.9 | 57.1 | 3.5 | 58.0 | 3.6 |

### 4.3 Generalization to More LVLMs

To validate the architectural robustness of TextScythe, we extend our evaluation to two distinct model families: Qwen2.5-VL-7B and LLaVA-OneVision-1.5-8B. As shown in Tables 4 and 5, TextScythe consistently outperforms competing pruning methods across both architectures under all pruning ratios, demonstrating effective cross-model generalization. On Qwen2.5-VL, TextScythe achieves high average performance retention rates of 97.6%, 94.0%, and 89.2% at pruning ratios of 66.7%, 77.8%, and 88.9%, respectively, outperforming the best baseline by 3.6 to 1.8 percentage points across these settings. On LLaVA-OneVision, it retains 97.7% accuracy under 70% pruning and 89.1% under 90% pruning, surpassing VisionZip by 2.0 to 5.5 percentage points. Notably, on semantically demanding benchmarks such as MME and POPE, TextScythe preserves nearly full-model accuracy even under aggressive compression, highlighting its ability to maintain critical semantic content. The consistent gains across both moderate and extreme pruning ratios underscore the effectiveness of our entropy-based instruction distillation in preserving task-relevant visual information. These results confirm that the instruction distillation paradigm generalizes effectively across diverse visual-language architectures, reinforcing its practicality as a plug-and-play acceleration module. Additional experimental results are provided in the Appendix A.4.

Table 4: Comparative Experiments on Qwen2.5-VL-7B.

| Methods | MMB | MME | POPE | SQA | $VQA_{Text}$ | Avg. |
|---|---|---|---|---|---|---|
| Upper Bound | 82.8 | 2304 | 86.1 | 84.7 | 84.8 | 100% |
| Qwen2.5-VL-7B | *Token Pruning Rate = 66.7% (Retain 33.3%)* | | | | | |
| FastV (ECCV24) | 75.7 | 2072 | 82.2 | 78.5 | 77.9 | 92.3% |
| PDrop (CVPR25) | 75.5 | 2043 | 81.8 | 78.0 | 77.2 | 91.6% |
| VisionZip (CVPR25) | 76.0 | 2097 | 82.9 | 78.8 | 78.3 | 92.9% |
| DART (EMNLP25) | 77.5 | 2106 | 83.1 | 77.6 | 78.6 | 93.2% |
| HoloV (NeurIPS25) | 78.3 | 2093 | 85.0 | 79.8 | 78.9 | 94.0% |
| TextScythe (Ours) | 81.4 | 2263 | 86.6 | 82.5 | 79.2 | 97.6% |
| Qwen2.5-VL-7B | *Token Pruning Rate = 77.8% (Retain 22.2%)* | | | | | |
| FastV (ECCV24) | 74.9 | 2036 | 80.7 | 78.0 | 69.5 | 89.3% |
| PDrop (CVPR25) | 75.0 | 2017 | 80.4 | 77.5 | 69.2 | 88.9% |
| VisionZip (CVPR25) | 75.7 | 2109 | 81.2 | 78.2 | 70.7 | 90.6% |
| DART (EMNLP25) | 76.1 | 2125 | 81.9 | 78.1 | 71.2 | 91.1% |
| HoloV (NeurIPS25) | 76.5 | 2043 | 82.3 | 79.8 | 70.3 | 92.4% |
| TextScythe (Ours) | 80.8 | 2177 | 85.5 | 81.2 | 70.1 | 94.0% |
| Qwen2.5-VL-7B | *Token Pruning Rate = 88.9% (Retain 11.1%)* | | | | | |
| FastV (ECCV24) | 71.2 | 1949 | 78.6 | 77.4 | 60.3 | 84.9% |
| PDrop (CVPR25) | 71.4 | 1920 | 77.0 | 76.9 | 60.5 | 84.2% |
| VisionZip (CVPR25) | 72.7 | 2006 | 77.5 | 77.8 | 61.9 | 85.9% |
| DART (EMNLP25) | 71.9 | 2042 | 77.9 | 76.9 | 61.7 | 85.9% |
| HoloV (NeurIPS25) | 72.4 | 2006 | 80.7 | 79.5 | 61.8 | 87.4% |
| TextScythe (Ours) | 76.2 | 2066 | 82.4 | 80.4 | 62.3 | 89.2% |

### 4.4 Efficiency Analysis

We conduct a comprehensive efficiency analysis to quantify the practical benefits of TextScythe. Following the evaluation protocol of related works, we measure total inference time, per-step latency, GPU

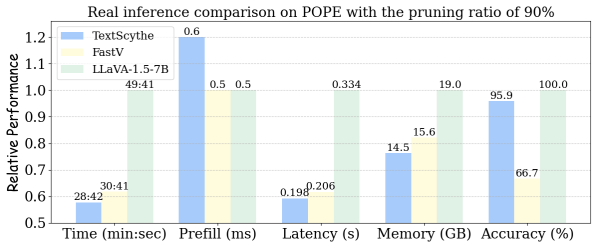


Figure 8: Efficiency Analysis.

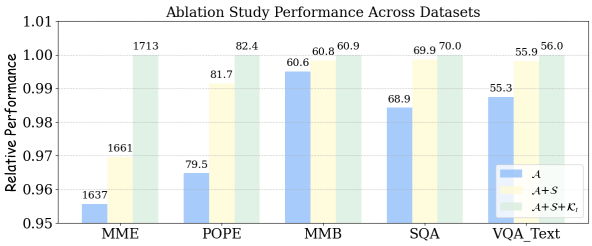


Figure 9: Impact of different components.

memory consumption, and accuracy on LLaVA-1.5 under a 90% pruning ratio. As shown in Fig. 8, TextScythe reduces total inference time by **42.1%** (from 49:41 to 28:42) and per-step latency by **40.7%** compared to the unpruned model, while retaining **95.9%** of the original accuracy.

Table 5: Comparative Experiments on LLaVA-OneVision.

| Methods | VizWiz | GQA | $VQA_{Text}$ | MME | MMB | POPE | Avg. |
|---|---|---|---|---|---|---|---|
| Upper Bound | 66.0 | 69.2 | 79.5 | 2271.3 | 85.3 | 88.5 | 100% |
| LLaVA-OneVision-1.5 | *Token Pruning Rate = 70.0% (Retain 30%)* | | | | | | |
| FastV (ECCV24) | 64.1 | 65.2 | 72.3 | 2019.5 | 79.6 | 70.4 | 90.7% |
| PDrop (CVPR25) | 62.5 | 64.0 | 70.2 | 1989.2 | 71.5 | 82.1 | 89.9% |
| VisionZip (CVPR25) | 64.7 | 65.9 | 73.2 | 2104.6 | 83.3 | 87.1 | 95.7% |
| TextScythe (Ours) | 65.0 | 66.9 | 76.2 | 2208.4 | 84.3 | 87.9 | 97.7% |
| LLaVA-OneVision-1.5 | *Token Pruning Rate = 90.0% (Retain 10%)* | | | | | | |
| FastV (ECCV24) | 60.9 | 61.3 | 56.5 | 1800.0 | 71.1 | 62.9 | 80.9% |
| PDrop (CVPR25) | 58.8 | 61.5 | 55.3 | 1829.7 | 70.5 | 69.9 | 81.6% |
| VisionZip (CVPR25) | 59.8 | 60.7 | 48.2 | 1980.3 | 73.3 | 79.3 | 83.6% |
| TextScythe (Ours) | 61.9 | 62.3 | 61.4 | 2008.5 | 76.6 | 84.1 | 89.1% |

A key finding is that TextScythe achieves **lower latency and memory usage than FastV** while delivering substantially higher accuracy (95.9 vs. 66.7 on POPE). This practical superiority stems from our architectural design: TextScythe performs a single, lightweight token selection *before* the LLM, preserving the dense attention pattern and maintaining full compatibility with highly optimized kernels like FlashAttention. In contrast, attention-based pruning methods like FastV require layer-wise sparsification and custom attention masking inside the transformer, which breaks kernel fusion, increases overhead, and limits hardware acceleration. These results confirm that TextScythe offers a favorable efficiency-accuracy trade-off suitable for real-world deployment.

### 4.5 Ablation Study and Analysis

To verify the effectiveness of each component within TextScythe, we conduct a series of ablation studies. All experiments are performed on LLaVA-1.5-7B with a visual token reduction rate of 90%. We systematically evaluate the contribution of three mechanisms: pruning using only the [CLS] attention ($\mathcal{A}$); $\mathcal{A}$ enhanced with similarity suppression between selected tokens ($\mathcal{A}+\mathcal{S}$), promoting visual diversity; and the complete framework that further incorporates key text token guidance ($\mathcal{A}+\mathcal{S}+\mathcal{K}_t$), which aligns visual selection with the instruction's semantics. As shown in Figure 9, the results demonstrate clear incremental gains. The similarity suppression mechanism ($\mathcal{A}+\mathcal{S}$) provides a consistent performance lift over the attention method ($\mathcal{A}$) across multiple benchmarks, validating the importance of reducing visual redundancy. The full model ($\mathcal{A}+\mathcal{S}+\mathcal{K}_t$) achieves the highest scores on all tasks, particularly on MME and POPE benchmarks,which confirming that combining visual diversity through similarity suppression with textual relevance through key token guidance is crucial for effective pruning.

## 5 Limitations

Performance depends on the quality of the cross-modal similarity signal. Extremely fine-grained tasks may require higher token budgets, and our proposed single-shot selection strategy does not yet incorporate temporal feedback for video modality. In the future, our work would includes adaptive budgets and multi-turn dialogue extensions.

## 6 Conclusion

We identify the *text-visual semantic misalignment* problem in attention-based visual token pruning for LVLMs and propose a new *instruction distillation* paradigm to address it. Our framework, **TextScythe**, first distills the user instruction into vision-critical text tokens using the Entropy-Ratio metric, then uses them to guide precise visual token selection while preserving global context through diversity-aware supplementation. Extensive experiments show TextScythe achieves state-of-the-art performance. It attains near-lossless compression, slightly surpassing unpruned accuracy at 77.8% pruning on LLaVA-1.5 model. The method generalizes across architectures and resolutions, and its pre-LLM pruning ensures compatibility with optimized kernels like FlashAttention.

## References


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

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 in resource-constrained environments. *arXiv preprint arXiv:2408.10945*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

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, and 8 others. 2025a. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

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

Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and 1 others. 2010. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342.

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

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. 2024. Matryoshka multimodal models. In *Workshop on Video-Language Models@ NeurIPS 2024*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024. 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.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, and 23 others. 2025. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *Preprint*, arXiv:2412.05271.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. *URL https://arxiv. org/abs/2307.08691*.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, and Xing Sun. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv:2306.13394*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. 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.

Wenbo Hu, Zi-Yi Dou, Liunian Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. 2024. Matryoshka query transformer for large vision-language models. *Advances in Neural Information Processing Systems*, 37:50168–50188.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Ahmadreza Jeddi, Negin Baghbanzadeh, Elham Dolatabadi, and Babak Taati. 2025. Similarity-aware token pruning: Your vlm but faster. *arXiv preprint arXiv:2503.11549*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Yanwei Li, Chengyao Wang, and Jiaya Jia. 2024b. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer.

Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. 2024c. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*.

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

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.

Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. *Springer International Publishing*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

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.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024c. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. 2024d. Multi-stage vision token dropping: Towards efficient multimodal large language model. *arXiv preprint arXiv:2411.10803*.

Xuyang Liu, Ziming Wang, Yuhang Han, Yingyao Wang, Jiale Yuan, Jun Song, Bo Zheng, Linfeng Zhang, Siteng Huang, and Honggang Chen. 2025a. Compression with global guidance: Towards training-free high-resolution mllms acceleration. *arXiv preprint arXiv:2501.05179*.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2025b. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pages 216–233. Springer.

Pan Lu, Swaroop Mishra, Tanglin 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. *Advances in Neural Information Processing Systems*, 35:2507–2521.

David Marr. 2010. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2024. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024a. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yiqi Wang, Wentao Chen, Xiaotian Han, Xudong Lin, Haiteng Zhao, Yongfei Liu, Bohan Zhai, Jianbo Yuan, Quanzeng You, and Hongxia Yang. 2024b. Exploring the reasoning abilities of multimodal large language models (mllms): A comprehensive survey on emerging trends in multimodal reasoning. *arXiv preprint arXiv:2401.06805*.

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

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. 2025b. Stop looking for important tokens in multimodal language models: Duplication matters more. *arXiv preprint arXiv:2502.11494*.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and 1 others. 2024. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. 2024a. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*.

Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, Minqi Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. 2024b. [cls] attention is all you need for training-free visual token pruning: Make vlm inference faster. *arXiv preprint arXiv:2412.01818*.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. 2025. Llava-mini: Efficient image and video large multimodal models with one vision token. *arXiv preprint arXiv:2501.03895*.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and 1 others. 2024c. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

Xin Zou, Di Lu, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Xu Zheng, Linfeng Zhang, and Xuming Hu. 2025a. Don't just chase" highlighted tokens" in mllms: Revisiting visual holistic context retention. *arXiv preprint arXiv:2510.02912*.

Xin Zou, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Kening Zheng, Sirui Huang, Junkai Chen, Peijie Jiang, Jia Liu, Chang Tang, and 1 others. 2025b. Look twice before you answer: Memory-space visual retracing for hallucination mitigation in multimodal large language models. In *Forty-second International Conference on Machine Learning*.

# Technical Appendices and Supplements

## A Appendix

In this appendix, we provide detailed information regarding the experimental setup, encompassing the datasets, model architectures, and comparison methods. Then, we offer a detailed analysis and discussion of the impact of hyperparameters on model performance.

### A.1 Detailed Experiment Settings

#### A.1.1 Datasets

We conducted experiments on several widely used visual understanding benchmarks. For image understanding task, we performed experiments on ten widely used benchmarks, including GQA (Hudson and Manning, 2019), MMBench (MMB) and MMB-CN (Liu et al., 2025b), MME (Fu et al., 2023), POPE (Li et al., 2023), VizWiz (Bigham et al., 2010), SQA (ScienceQA) (Lu et al., 2022), $\text{VQA}_{\text{V2}}$ (VQA V2) (Goyal et al., 2017) and $\text{VQA}_{\text{Text}}$ (TextVQA) (Singh et al., 2019)

**GQA.** (Hudson and Manning, 2019) The GQA benchmark is composed of three parts: scene graphs, questions, and images. The image part contains images, as well as the spatial features of images and the features of all objects in images. The questions in GQA are designed to test the understanding of visual scenes and the ability to reason about different aspects of an image.

**MMBench.** (Liu et al., 2025b) The MMBench benchmark comprehensively evaluates the model's overall performance across multiple dimensions. It includes three levels of ability dimensions. The first level (L-1) consists of two main abilities, perception and reasoning. The second level (L-2) expands based on the first level, including six sub-abilities. The third level (L-3) further refines the second level, encompassing 20 specific ability dimensions. This hierarchical structure enables a granular and comprehensive evaluation of the model's various capabilities.

**MME.** (Fu et al., 2023) The MME benchmark is also a comprehensive benchmark meticulously designed to thoroughly evaluate various aspects of a model's performance. It consists of 14 subtasks that specifically aim to evaluate both the model's perceptual and cognitive abilities. By utilizing manually constructed instruction-answer pairs and concise instruction design, it effectively mitigates issues such as data leakage and unfair evaluation of model performance.

**POPE.** (Li et al., 2023) The POPE benchmark is primarily used to evaluate the degree of Object Hallucination in models. It reformulates hallucination evaluation by requiring the model to answer a series of specific binary questions regarding the presence of objects in images. Accuracy, Recall, Precision, and F1 Score are effectively employed as reliable evaluation metrics to precisely measure the model's hallucination level under three different sampling strategies.

**ScienceQA.** (Lu et al., 2022) The ScienceQA benchmark covers a rich diversity of domains, including natural science, language science, and social science. Within each subject, questions are categorized first by the topic, then by the category, and finally by the skill. This hierarchical categorization results in 26 topics, 127 categories, and 379 skills, providing a comprehensive and diverse range of scientific questions. It provides a comprehensive evaluation of a model's capabilities in multimodal understanding, multi-step reasoning, and interpretability.

**VQA-v2.** (Goyal et al., 2017) The VQA-v2 benchmark evaluates the model's visual percep-

tion capabilities through open-ended questions. It consists of 265,016 images, covering a wide variety of real-world scenes and objects, providing rich visual contexts for the questions. For each question, there are 10 ground truth answers provided by human annotators, which allows for a comprehensive evaluation of the performance of different models in answering the questions accurately.

**TextVQA.** (Singh et al., 2019) The TextVQA benchmark focuses on the comprehensive integration of diverse text information within images. It meticulously evaluates the model's text understanding and reasoning abilities through a series of visual question-answering tasks with rich textual information. Models need to not only understand the visual content of the images but also be able to read and reason about the text within the images to answer the questions accurately.

### A.1.2 Models

We evaluate TextScythe using various open-source MLLMs. For image understanding tasks, experiments are conducted on the LLaVA family, including LLaVA-1.5-7B[1] (Liu et al., 2024a) and LLaVA-Next-7B[2] (Liu et al., 2024b), with the latter used to validate performance on high-resolution images. Furthermore, we validate our method on other advanced model Qwen2.5-VL-7B (Bai et al., 2025a). For video understanding tasks, we use Video-LLaVA (Lin et al., 2023) as the baseline model. following the settings reported in their paper to ensure a fair comparison.

### A.1.3 Baselines

We analyze multiple representative methods for accelerating multi-modal language models (MLLMs) through token reduction. These methods share the goal of improving efficiency by reducing redundant tokens, yet differ in their strategies, such as token merging, pruning, or adaptive allocation.

**ToMe** (Bolya et al., 2022) merges similar tokens in visual transformer layers through lightweight matching techniques, achieving acceleration without requiring additional training.

**FastV** (Chen et al., 2024) focuses on early-stage token pruning by leveraging attention maps, effectively reducing computational overhead in the initial layers.

**SparseVLM** (Zhang et al., 2024c) ranks token importance using cross-modal attention and introduces adaptive sparsity ratios, complemented by a novel token recycling mechanism.

**HiRED** (Arif et al., 2024) allocates token budgets across image partitions based on CLS token attention, followed by the selection of the most informative tokens within each partition, ensuring spatially aware token reduction.

**LLaVA-PruMerge** (Shang et al., 2024) combines pruning and merging strategies by dynamically removing less important tokens using sparse CLS-visual attention and clustering retained tokens based on key similarity.

**PDrop** (Xing et al., 2024) adopts a progressive token-dropping strategy across model stages, forming a pyramid-like token structure that balances efficiency and performance.

**FasterVLM** (Zhang et al., 2024b) evaluates token importance via CLS attention in the encoder and performs pruning before interaction with the language model, streamlining the overall process.

**MustDrop** (Liu et al., 2024d) integrates multiple strategies, including spatial merging, text-guided pruning, and output-aware cache policies, to reduce tokens across various stages.

**GlobalCom**[2] (Liu et al., 2025a) introduces a hierarchical approach by coordinating thumbnail tokens to allocate retention ratios for high-resolution crops while preserving local details.

**DART** (Wen et al., 2025b) introduces a duplication-aware token reduction method that selects a small subset of pivot tokens, calculates cosine similarity between pivot tokens and remaining tokens, retains those with the lowest duplication to pivots, achieving significant acceleration while maintaining performance and good compatibility with efficient attention operators.

These methods collectively highlight diverse approaches to token reduction, ranging from attention-based pruning to adaptive merging, offering complementary solutions for accelerating MLLMs.

### A.1.4 Implementation Details

All of our experiments are conducted on Nvidia A800-80G GPU. The implementation was carried out in Python 3.10, utilizing PyTorch 2.1.2, and CUDA 11.8. All baseline settings follow the original paper. Our hyperparameter design is $\alpha$=0.7 .

[1] https://huggingface.co/liuhaotian/llava-v1.5-7b

[2] https://huggingface.co/liuhaotian/llava-v1.6-vicuna-7b

## A.2 Related Work

### A.2.1 Multimodal large language models

Large Language Models (LLMs) (Bai et al., 2023; Jiang et al., 2023; Ouyang et al., 2022; Touvron et al., 2023) have recently achieved remarkable success, leading to a growing trend of extending their powerful reasoning capabilities to multimodal understanding tasks, ultimately giving rise to Multimodal Large Language Models (MLLMs) (Liu et al., 2024c; Li et al., 2024a; Wang et al., 2024a; Bai et al., 2025b; Chen et al., 2025; Zhu et al., 2025; Zou et al., 2025b). These models typically encode visual inputs into tokens to fully leverage LLMs' capabilities. While enabling visual perception, this approach introduces substantial computational overhead from long visual token sequences. For example, LLaVA-1.5 (Liu et al., 2024a) converts a $336 \times 336$ image into 576 tokens, while its high-resolution variant LLaVA-NeXT (Liu et al., 2024b) generates 2,880 tokens from double-resolution images. In video understanding scenarios, models like LongVA (Zhang et al., 2024a) can produce ultra-long sequences exceeding 200K visual tokens. Thus, it is crucial to accelerate MLLM inference.

### A.2.2 Visual Token Compression

One effective approach to optimizing MLLM inference involves reducing the predominantly visual tokens in input sequences. Compared to text dense with information, visual signals exhibit greater spatial redundancy (Marr, 2010). While some works attempt visual token compression through vision-text prefusion (Li et al., 2024b; Hu et al., 2024; Cai et al., 2024; Zhang et al., 2025), these methods require architectural modifications and additional training, thereby increasing computational costs. Alternative training-free approaches, known as token pruning, remove redundant visual tokens during inference. FastV (Chen et al., 2024) first identified the redundancy in LVLMs and proposed pruning low-attention visual tokens after the second layer of the language model. SparseVLM (Zhang et al., 2024c) eliminates text prompt interference and employs more accurate text attention for progressive visual token sparsification. However, such text-visual attention-based methods suffer from text-visual semantic misalignment issues (Zhang et al., 2024b; Wen et al., 2025a) that compromise pruning accuracy, and they remain incompatible with efficient attention implementations like FlashAttention (Dao et al., 2022; Dao, 2023). Other studies (Wen et al., 2025b; Alvar et al., 2025; Jeddi et al., 2025) prune tokens based on inter-token feature similarity, but ignore the critical relevance between visual tokens and user instructions, leading to suboptimal performance. However, our proposed TextScythe addresses these limitations by simultaneously optimizing instruction relevance and token distinctiveness for more effective visual pruning while maintaining hardware acceleration compatibility.

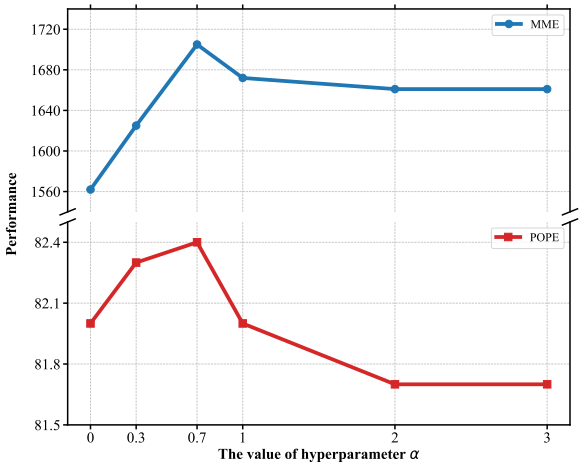


Figure 10: Impact of hyperparamete $\alpha$.

## A.3 Impact of hyperparamete

We analyze the impact of the text token selection hyperparameter $\alpha$ on model performance. As shown in Fig. 10, performance on both MME and POPE improves as $\alpha$ increases from 0 to 0.7, reaching an optimum. We hypothesize that when $\alpha$ is too small, text token selection becomes overly lenient, allowing non-visual or irrelevant words to be incorrectly identified as key tokens, which in turn misguides visual token selection. Conversely, when $\alpha$ exceeds 0.7, selection becomes overly strict, potentially excluding legitimate key text tokens that are essential for capturing instruction-relevant visual content, leading to a gradual performance decline. The peak at $\alpha = 0.7$ demonstrates that our adaptive thresholding mechanism effectively balances inclusivity and precision in text token selection, ensuring that only the most vision-critical tokens guide pruning. This result confirms the importance of properly calibrating $\alpha$ to maximize the accuracy of instruction distillation.

Table 6: Performance comparison on text-dense benchmarks with different token pruning ratios.

| Methods | OCR Bench | Chart QA | Chinese OCRbench | Avg. |
|---|---|---|---|---|
| Upper Bound | 297 | 224 | 24 | 100% |
| LLaVA1.5-7B | *Token Pruning Rate = 66.7%* | | | |
| FastV (ECCV24) | 190 | 202 | 23 | 83.3% |
| PDrop (CVPR25) | 290 | 209 | 24 | 97.0% |
| TextScythe (Ours) | 296 | 214 | 26 | 101.2% |
| LLaVA1.5-7B | *Token Pruning Rate = 77.8%* | | | |
| FastV (ECCV24) | 191 | 183 | 22 | 79.2% |
| PDrop (CVPR25) | 287 | 190 | 24 | 92.4% |
| TextScythe (Ours) | 295 | 210 | 25 | 99.1% |
| LLaVA1.5-7B | *Token Pruning Rate = 88.9%* | | | |
| FastV (ECCV24) | 191 | 155 | 15 | 65.3% |
| PDrop (CVPR25) | 250 | 162 | 17 | 76.8% |
| TextScythe (Ours) | 287 | 172 | 18 | 82.8% |

### A.4 More Experient Results

#### A.4.1 Performance on Text-dense Benchmarks

As shown in Table 6, we further evaluate TextScythe on text-dense benchmarks that demand precise retention of visual tokens containing textual information. TextScythe consistently outperforms FastV and PDrop across all pruning ratios, demonstrating its ability to preserve semantically critical visual content even when fine-grained textual details are essential. Remarkably, at 66.7% pruning, TextScythe achieves an average performance of **101.2%**, slightly exceeding the unpruned upper bound—indicating that filtering out irrelevant visual tokens can enhance the model's focus on text-relevant regions. Even under extreme 88.9% pruning, it retains **82.8%** of the original performance, significantly surpassing the best baseline. These results confirm that TextScythe's instruction-distillation mechanism effectively identifies and retains tokens crucial for text recognition and dense visual understanding.

#### A.4.2 More Experiments on Qwen2.5-VL Models

We supplemented extra experiments on Qwen2.5-VL, as shown in Table 7. Our method achieves the best performance across all pruning ratios compared with other state-of-the-art methods.

## B Ethics Statement

This work presents a method for improving the computational efficiency of vision-language models through token pruning. We recognize the following ethical considerations:

**Positive Impacts:** Our method can reduce the computational cost and energy consumption of large AI models, contributing to more environmentally sustainable AI deployment. This could make advanced AI capabilities more accessible in resource-constrained environments.

**Potential Risk:** While token pruning generally preserves model performance, aggressive pruning might potentially amplify biases or affect model fairness by disproportionately removing information about underrepresented visual concepts. However, our experiments show that TextScythe maintains robust performance across diverse benchmarks.

**Data Usage:** Our research uses publicly available benchmarks and models. All datasets employed in this study are widely used in the research community for non-commercial purposes.

**Broader Implications:** We believe the efficiency improvements offered by our method align with responsible AI development goals by reducing the computational barrier to using advanced multimodal AI systems

## C Reproducibility Statement

To ensure the reproducibility of our work, we provide the following:

**Code Availability:** The implementation of TextScythe will be made publicly available upon publication.

**Experimental Details:**

- Complete hyperparameter settings for all experiments are provided in Appendix A.1.4.

- The detailed method implementation process is described in Section 3.

- The specific versions of all baseline methods we compared against are clearly cited.

**Datasets:** All datasets used in this study are publicly available.

**Models:** Our experiments use publicly available model checkpoints.

**Computational Resources:** We report the specific hardware configurations and computational requirements in Appendix A.1.4. All experiments can be reproduced with similar GPU resources.

Table 7: Comprehensive Comparative Experiments on Qwen2.5-VL-7B across multiple benchmarks.

| Method | GQA | MMB | MME | POPE | SQA | $VQA_{V2}$ | $VQA_{Text}$ | VizWiz | Average |
|---|---|---|---|---|---|---|---|---|---|
| *Original Model (Retain 100% Tokens)* | | | | | | | | | |
| Qwen2.5-VL-7B | 65.2 | 82.8 | 2304 | 86.1 | 84.7 | 92.3 | 84.8 | 68.3 | 100% |
| *Token Pruning Rate = 66.7% (Retain 33.3% Tokens)* | | | | | | | | | |
| +FastV (ECCV24) | 61.0 | 75.7 | 2072 | 82.2 | 78.5 | 86.5 | 77.9 | 64.1 | 92.8% |
| +PDrop (CVPR25) | 60.7 | 75.5 | 2043 | 81.8 | 78.0 | 86.6 | 77.2 | 63.7 | 92.3% |
| +VisionZip (CVPR25) | 62.5 | 76.0 | 2097 | 82.9 | 78.8 | 87.0 | 78.3 | 65.0 | 93.7% |
| +DART (EMNLP25) | 63.2 | 77.5 | 2106 | 83.1 | 77.6 | 85.9 | 78.6 | 65.4 | 93.9% |
| +TextScythe (Ours) | **63.9** | **81.4** | **2263** | **86.6** | **82.5** | **87.6** | **79.2** | **65.1** | **97.0%** |
| *Token Pruning Rate = 77.8% (Retain 22.2% Tokens)* | | | | | | | | | |
| +FastV (ECCV24) | 60.5 | 74.9 | 2036 | 80.7 | 78.0 | 82.3 | 69.5 | 63.5 | 90.2% |
| +PDrop (CVPR25) | 60.2 | 75.0 | 2017 | 80.4 | 77.5 | 82.1 | 69.2 | 64.0 | 89.6% |
| +VisionZip (CVPR25) | 61.8 | 75.7 | 2109 | 81.2 | 78.2 | 83.0 | 70.7 | 65.0 | 91.4% |
| +DART (EMNLP25) | 62.0 | 76.1 | 2125 | 81.9 | 78.1 | 83.2 | 71.2 | 63.5 | 91.7% |
| +TextScythe (Ours) | **62.4** | **80.8** | **2177** | **85.5** | **81.2** | **85.8** | **70.1** | **64.7** | **94.2%** |
| *Token Pruning Rate = 88.9% (Retain 11.1% Tokens)* | | | | | | | | | |
| +FastV (ECCV24) | 57.2 | 71.2 | 1949 | 78.6 | 77.4 | 81.0 | 60.3 | 60.5 | 86.1% |
| +PDrop (CVPR25) | 56.3 | 71.4 | 1920 | 77.0 | 76.9 | 81.5 | 60.5 | 60.3 | 85.5% |
| +VisionZip (CVPR25) | 58.0 | 72.7 | 2006 | 77.5 | 77.8 | 81.7 | 61.9 | 61.5 | 87.2% |
| +DART (EMNLP25) | 58.5 | 71.9 | 2042 | 77.9 | 76.9 | 81.3 | 61.7 | 61.2 | 87.1% |
| +TextScythe (Ours) | **60.5** | **76.2** | **2066** | **82.4** | **80.4** | **83.1** | **62.3** | **62.5** | **90.0%** |

## D The Use of Large Language Models (LLMs)

In preparing this manuscript, we utilized DeepSeek-R1 as a writing and editing assistant. Its role was limited to enhancing the clarity and fluency of the English in various sections. All scientific ideas, research methodology, experimental design, result analysis, and technical contributions are solely the product of the human authors. DeepSeek was not involved in any aspect of research conception, algorithm design, data interpretation, or validation of mathematical formulations, theoretical analyses, and experimental results.