MIND THE EVIDENCE: LLM-ALIGNED VISUAL TOKEN COMPRESSION FOR VLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models increasingly rely on large numbers of visual tokens, making visual token compression essential for efficient inference. However, existing token pruning methods typically estimate token importance from the vision encoder or from attention/relevance patterns, without explicitly considering how visual information is transformed before reaching the language model. This creates a fundamental mismatch: a token that appears important in the visual representation may contribute little to the representation actually processed by the LLM. We introduce LAVIC, a training-free framework for compressing visual tokens from the perspective of the LLM-facing representation. LAVIC measures how distinguishable local visual variations remain in the LLM-facing representation, using this distinguishability as a signal of the visual evidence exposed to the downstream language model. Based on this evidence, LAVIC preserves informative tokens, groups redundant evidence around query-relevant anchors, and adaptively allocates the limited token budget according to the concentration of the evidence. Across multiple vision-language models and benchmarks, LAVIC consistently preserves model performance under aggressive token compression while substantially reducing visual tokens and inference cost. On LLaVA-v1.5, LAVIC retains 96.4% of the full-token performance using only 60 of 576 visual tokens, while reducing prefill latency from 58.70 ms to 28.95 ms. At 128 and 64 tokens, LAVIC also maintains strong performance and consistently outperforms competing compression methods. These results suggest that visual token compression can be better understood as preserving LLM-facing visual evidence rather than simply selecting visually salient tokens.

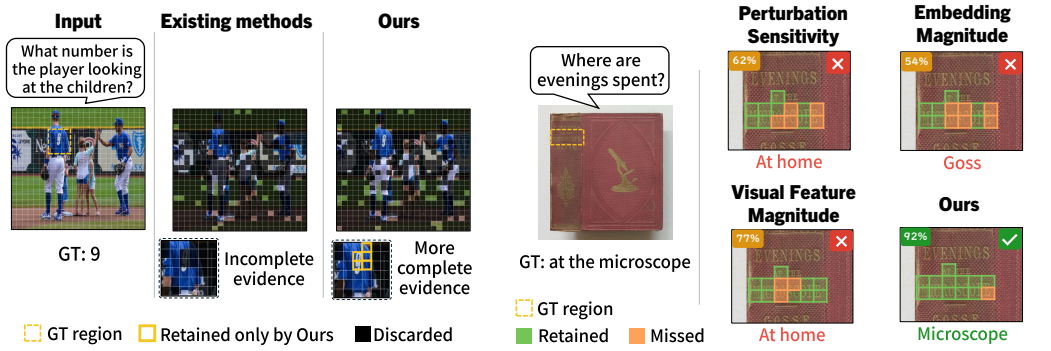
1 INTRODUCTION

Vision-language models (VLMs) increasingly rely on large numbers of visual tokens to represent images and videos. These visual tokens are directly processed by the language model, so their number can substantially affect both computation and inference latency. This cost becomes particularly important for high-resolution images, multi-image inputs, and videos, where visual sequences can contain hundreds or thousands of tokens. As a result, visual token compression has emerged as an effective way to improve the efficiency of VLM inference without retraining the underlying model.

A natural approach is to identify visual tokens that are less important and remove them. Existing methods have explored a variety of signals for this purpose, including visual feature magnitude, attention, text-vision relevance, redundancy, diversity, sensitivity, and entropy. While these criteria can be effective, they largely treat compression as a *token-importance problem*: given a visual sequence, determine which individual tokens should be retained or discarded. However, the importance of a visual token in the vision representation does not necessarily indicate how much useful visual evidence it provides to the downstream language model.

The key reason is that VLMs do not directly expose the original vision representation to the language model. Instead, visual features are transformed by a vision-to-language interface before they are processed by the LLM. Consequently, two visual tokens with similar importance according to a vision-side criterion may provide very different information to the LLM, while a token that appears less important in the vision representation may still preserve evidence that is important for answering the query. As illustrated in Fig. 1, conventional vision-side importance criteria can retain

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.



(a) Matched Token Count, Different Evidence (b) Vision-Side Importance Can Miss Answer Evidence

Figure 1: **Vision-side importance does not always reflect LLM-aligned evidence.** Even at the same token count, conventional compression criteria may fail to retain evidence relevant to the correct response. LAVIC utilizes LLM-aligned visual evidence as the basis for visual token compression. Yellow dotted boxes are shown only for visualization and are not used during inference.

the same number of tokens while preserving different amounts of evidence relevant to the correct response. This motivates a different question: *Instead of asking which visual tokens are important, can we identify and preserve the visual evidence that is most clearly expressed in the representation consumed by the LLM?*

We refer to this information as *LLM-aligned visual evidence*, namely visual information that supports downstream LLM processing after compression. This perspective shifts visual token compression from selecting important tokens to *preserving LLM-aligned evidence under a limited token budget*. Importantly, LLM-aligned visual evidence is not intended to represent an exact measure of semantic importance or answer correctness. Rather, it captures how strongly local visual distinctions are expressed in the LLM-facing representation, providing a practical signal for deciding which information should remain explicit after compression.

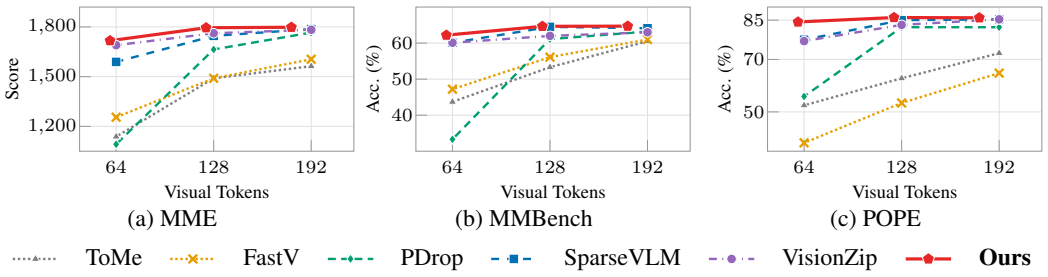
Based on this perspective, we introduce **LAVIC**, a training-free and plug-and-play framework for LLM-aligned visual compression. LAVIC uses LLM-aligned visual evidence throughout the compression process. First, before the visual sequence enters the LLM, LAVIC measures how distinguishable the local variations of each visual token are from one another after projection into the LLM-facing representation. Tokens with more distinguishable projected variations are considered to carry stronger LLM-facing evidence and are prioritized for explicit retention. This provides a compression criterion that directly operates on the representation seen by the downstream language model rather than only on the vision-side representation.

However, retaining only individually important tokens can still lose useful evidence when the token budget becomes highly constrained. A query may require multiple pieces of visual information, and tokens that are removed during compression may contain different types of query-relevant evidence. Simply discarding these tokens or summarizing them jointly can mix distinct evidence together. LAVIC therefore organizes removal candidates according to *query anchors* and carries their information forward through separate query-anchor-specific summaries. In this way, compression can reduce redundancy while retaining distinctions among different evidence relevant to the query.

Finally, the amount of visual information required by the LLM is not necessarily uniform across an input or across layers. Some inputs contain concentrated query-relevant evidence, while others require broader visual coverage. LAVIC therefore adapts the visual capacity according to the spread of query-conditioned evidence rather than imposing a fixed allocation. This allows the limited token budget to better match the amount of visual evidence required by the LLM throughout compression.

These three operations follow a single principle: *preserve the evidence that remains meaningful in the LLM-facing space, rather than optimizing token selection independently at each compression stage.* The resulting framework requires no additional training and can be applied as a plug-and-play compression module to existing VLMs.

We evaluate LAVIC across diverse VLM architectures and visual input settings, including standard-resolution images, high-resolution images, multi-image inputs, and videos, under a wide range of



(a) MME (b) MMBench (c) POPE

Figure 2: **Performance across visual-token budgets on LLaVA-v1.5-7B (Liu et al., 2024a).** LAVIC maintains strong performance across diverse benchmarks, with its advantages becoming more pronounced under aggressive compression.

visual-token budgets. As shown in Fig. 2, LAVIC consistently maintains strong downstream performance as the visual sequence is compressed, with its advantage becoming particularly pronounced under aggressive compression. On LLaVA-v1.5-7B, for example, LAVIC retains **96.4% of the uncompressed aggregate performance using only 60 of 576 visual tokens**, while reducing prefill latency by **50.7%** at a representative compression setting. Additional evidence-level analyses further show that the performance gains are closely associated with preserving LLM-aligned visual evidence.

Our main contributions are summarized as follows:

- **From token importance to LLM-aligned visual evidence.** We identify a mismatch between vision-side token importance and the visual evidence expressed in the representation consumed by the downstream LLM, and formulate visual token compression from the perspective of LLM-aligned visual evidence.
- **LLM-aligned visual compression.** We introduce LAVIC, a training-free and plug-and-play framework that preserves LLM-aligned evidence through LLM-facing filtering based on the distinguishability of projected local variations, query-anchor-specific evidence summarization, and evidence-adaptive visual capacity.
- **Broad evaluation and evidence-level analysis.** We evaluate LAVIC across diverse VLMs, visual input settings, and compression levels, demonstrating favorable performance-efficiency trade-offs and providing evidence that LLM-aligned visual evidence is an effective basis for aggressive visual token compression.

2 Related Work

Token Compression in Vision Transformers. A large body of work has studied token pruning, merging, and dynamic sparsification to reduce the computational cost of Vision Transformers. DynamicViT (Rao et al., 2021) and A-ViT (Yin et al., 2022) dynamically remove less important tokens based on the input, while EViT (Liang et al., 2022) preserves attentive tokens while fusing less informative ones. ToMe (Bolya et al., 2023) reduces sequence length by merging tokens with similar features. Subsequent methods, including Token Fusion (Kim et al., 2024) and MCTF (Lee et al., 2024), further refine token reduction using attention-, similarity-, and structure-aware mechanisms. These works laid the foundation for visual token reduction, which was later extended to VLMs.

Visual Token Compression for VLMs. In VLMs, visual tokens are directly processed by the LLM, making long visual sequences costly and motivating a growing body of work on visual token compression. Recent VLMs span increasingly diverse architectures and visual input regimes. Across these models, visual inputs range from fixed-resolution images (Liu et al., 2024a; Bai et al., 2023) to high-resolution images (Liu et al., 2024b; Li et al., 2025), multi-image inputs (Chen et al., 2024b; Li et al., 2024; Bai et al., 2025), and videos (Lin et al., 2024), often producing substantially longer and more structurally diverse visual sequences.

To handle these increasingly long and heterogeneous visual sequences, existing methods employ a wide range of compression signals and strategies, including attention, text-vision relevance, redundancy, diversity, sensitivity, and entropy. FastV (Chen et al., 2024a) and SparseVLM (Zhang

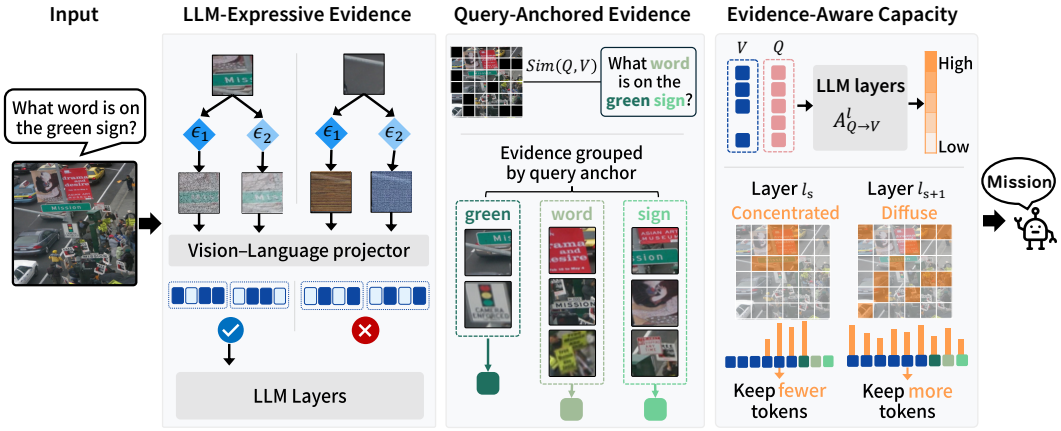


Figure 3: **Overview of LAVIC.** LAVIC utilizes LLM-aligned visual evidence throughout compression by prioritizing evidence that remains distinctly expressed in the LLM-facing space, carrying information from removal candidates through query-anchor-specific summaries, and adapting visual capacity according to the spread of query-conditioned evidence.

et al., 2025b) leverage attention or text-vision interactions, while VisionZip (Yang et al., 2025), DART (Wen et al., 2025), DivPrune (Alvar et al., 2025), and CDPruner (Zhang et al., 2025a) use attention structure, redundancy, or diversity as signals for visual token selection. PyramidDrop (Xing et al., 2025) and ATP-LLaVA (Ye et al., 2025) consider layer-wise or instance-adaptive compression, while CoViPAL (Tang et al., 2025) uses contextualized layer-wise pruning. PruMerge (Shang et al., 2025), VLM-Pruner (Wu et al., 2026), and VisionTrim (Yu et al., 2026) incorporate token merging or information fusion alongside pruning. More recently, methods such as ZOO-Prune (Kim et al., 2026) and EntropyPrune (Wang et al., 2026) have explored perturbation sensitivity and entropy-based signals for visual token compression.

These studies have substantially advanced visual token compression with diverse criteria for reducing visual sequences. However, despite their different signals and strategies, they largely treat compression as a token-level decision problem rather than focusing on the visual evidence that supports downstream LLM processing. We instead formulate visual token compression around LLM-aligned visual evidence, focusing on how this evidence is preserved and utilized throughout compression.

3 LAVIC: LLM-Aligned Visual Compression

Given an image I and a query Q, the vision encoder produces N visual tokens, $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{N}$. Under a limited visual-token budget, LAVIC utilizes LLM-aligned visual evidence throughout compression. As illustrated in Fig. 3, LAVIC prioritizes evidence in the LLM-facing space, carries information from removal candidates through query-anchor-specific summaries, and adjusts visual capacity according to the spread of query-conditioned evidence.

3.1 LLM-Expressive Evidence

Before the visual sequence enters the LLM, LAVIC filters it to prioritize LLM-aligned visual evidence. Rather than relying on vision-side saliency, we evaluate visual tokens in the representation seen by the LLM. Specifically, we generate local variations of each visual token and project them into the LLM-facing space, then measure how distinguishable the resulting representations are from one another. Tokens with more distinguishable variations are prioritized for explicit retention.

Let $g(\cdot)$ denote the visual-to-LLM projector. For each visual token $\mathbf{v}_i$, we construct M local variants by adding zero-mean isotropic Gaussian noise. This provides a simple way to sample local perturbations without favoring any particular direction. Since the criterion only requires comparing the projected representations of local variations, other perturbation distributions can also be used.

$$\mathbf{z}_{i,m} = g(\mathbf{v}_i + \boldsymbol{\epsilon}_{i,m}), \quad \boldsymbol{\epsilon}_{i,m} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}), \qquad s_i = \frac{1}{M^2 D} \sum_{m<m'} \|\mathbf{z}_{i,m} - \mathbf{z}_{i,m'}\|_2^2 . \tag{1}$$

Here, i indexes visual tokens, m and m' index their local variants, and D denotes the LLM-facing dimension. A larger s_i indicates that the local variations of token i are more distinguishable from one another after projection. LAVIC retains the tokens with the largest s_i to form the visual sequence $\mathbf{V}_0$ that enters the LLM. This yields a visual sequence concentrated in LLM-expressive evidence. We use $M = 2$ and $\sigma = 0.002$ as the default settings.

3.2 Query-Anchored Evidence

After the initial LLM-facing filtering, subsequent compression within the LLM may still remove visual information relevant to the query. When a query requires multiple pieces of visual information, removal candidates may carry different query-relevant evidence. Summarizing them jointly can obscure these distinctions, so LAVIC groups them by query anchor and summarizes each group separately.

To identify query anchors, let $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_T]$ denote the query-token embeddings. Before entering the LLM layers, we measure the association of each query token with the filtered visual sequence $\mathbf{V}_0$:

$$\eta_t = \frac{1}{|\mathbf{V}_0|} \sum_{\mathbf{v}_i \in \mathbf{V}_0} \mathrm{Softmax}_t\big(g(\mathbf{v}_i)^\top \mathbf{Q}\big), \qquad \mathcal{Q}_A = \{\mathbf{q}_t \mid \eta_t > 1/T\}. \tag{2}$$

Here, $1/T$ denotes the uniform association level across query tokens. Query tokens with above-uniform visual association form $\mathcal{Q}_A$ and serve as anchors for organizing removal candidates.

At pruning layer l, LAVIC assigns each visual token a relevance score $r_i^{(l)}$ using attention from the selected query anchors. We define this score and the query-anchor grouping as:

$$r_i^{(l)} = \frac{1}{|\mathcal{Q}_A|} \sum_{t:\mathbf{q}_t \in \mathcal{Q}_A} A_{t\to i}^{(l)}, \qquad \mathcal{G}_t^{(l)} = \left\{ i \in \mathcal{D}^{(l)} \;\middle|\; t = \arg\max_{u:\mathbf{q}_u \in \mathcal{Q}_A} A_{u\to i}^{(l)} \right\}, \qquad \mathbf{q}_t \in \mathcal{Q}_A. \tag{3}$$

Here, i indexes visual tokens at layer l, and $A_{t\to i}^{(l)}$ denotes the head-averaged attention from query anchor $\mathbf{q}_t$ to visual token i. This reuses the attention already computed by the LLM and therefore requires no additional attention pass. Under the layer-l token budget, $r_i^{(l)}$ determines which visual tokens are retained, while the remaining tokens form the removal candidate set $\mathcal{D}^{(l)}$. The removal candidates are then assigned to their most strongly associated query anchors, forming the groups $\mathcal{G}_t^{(l)}$. The grouped removal candidates are then summarized using relevance-weighted averages of their visual token representations at layer l, with $r_i^{(l)}$ as weights. The resulting query-anchor-specific summaries are passed to subsequent LLM layers together with the retained tokens, allowing query-relevant evidence associated with different anchors to remain distinct.

3.3 Evidence-Aware Capacity

The relevance scores $r_i^{(l)}$ also indicate how broadly query-relevant evidence is distributed across visual tokens. When relevance is concentrated in a few tokens, stronger compression may be sufficient, whereas broadly distributed relevance requires more visual capacity.

LAVIC therefore adapts the layer-wise visual capacity according to the spread of $r_i^{(l)}$. To quantify this spread, we normalize the relevance scores and compute their normalized Shannon entropy:

$$p_i^{(l)} = \frac{r_i^{(l)}}{\sum_k r_k^{(l)}}, \qquad \mathrm{spread}^{(l)} = \frac{-\sum_i p_i^{(l)} \log p_i^{(l)}}{\log N^{(l)}}. \tag{4}$$

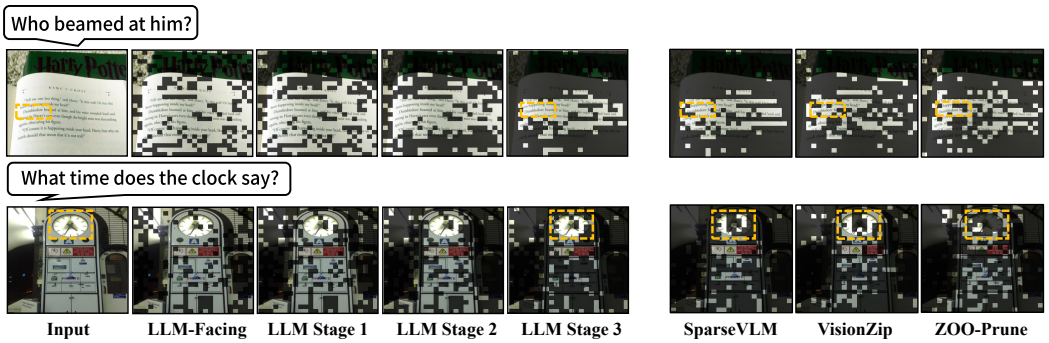
Here, $p_i^{(l)}$ denotes the normalized relevance of visual token i at layer l, and $N^{(l)}$ denotes the number of visual tokens at that layer. A lower $\mathrm{spread}^{(l)}$ indicates more concentrated query relevance, while a higher value indicates more distributed relevance. LAVIC adjusts the layer-wise compression scale according to this spread:

$$\mathrm{scale}^{(l)} = \alpha + \beta \left(1 - \mathrm{spread}^{(l)}\right), \tag{5}$$

Here, α sets the base compression scale and β controls its adaptation to relevance spread. A lower spread increases the compression scale, leading to stronger compression, whereas a higher spread

Table 1: **Main results on LLaVA-v1.5-7B and LLaVA-NeXT-7B under comparable visual-token budgets.** Acc. denotes the average relative performance across benchmarks.

LLaVA-v1.5-7B (Liu et al., 2024a)								**LLaVA-NeXT-7B** (Liu et al., 2024b)			
Method	**Tokens**	**MMB**	**MME**	**POPE**	**SQA**	**VQA^{Text}**	**Acc. (%)**	**Method**	**MME**	**SQA**	**Acc. (%)**
Upper bound, 576 tokens (100%)								*Upper bound, 2880 tokens (100%)*			
Vanilla	576	64.6	1864	85.9	69.5	58.3	100	Vanilla	1842	70.2	100
Target 192 Tokens (↓ 66.7%)								*Target 640 Tokens (↓ 77.8%)*			
PDrop CVPR'25	192	63.2	1766	82.3	70.2	56.1	97.1 (↓ 2.9)	SparseVLM ICML'25	1772	67.7	96.3 (↓ 3.7)
SparseVLM ICML'25	192	64.1	1787	85.3	68.7	57.8	98.5 (↓ 1.5)	VisionZip CVPR'25	1787	68.1	97.0 (↓ 3.0)
VisionZip CVPR'25	192	63.0	1783	85.3	68.9	57.3	98.0 (↓ 2.0)	DivPrune CVPR'25	1773	67.8	96.4 (↓ 3.6)
DivPrune CVPR'25	192	62.0	1771	87.7	68.7	56.6	97.8 (↓ 2.2)	VisionTrim ICLR'26	1825	70.7	99.9 (↓ 0.1)
ZOO-Prune CVPR'26	192	62.9	1782	87.2	69.2	57.3	98.5 (↓ 1.5)	ZOO-Prune CVPR'26	1816	68.0	97.7 (↓ 2.3)
Ours	**179** (↓ 68.9%)	**64.7**	**1798**	**85.9**	**69.8**	**57.5**	**99.1** (↓ 0.9)	**Ours**	**1835**	**73.4**	**102.1** (↑ 2.1)
Target 128 Tokens (↓ 77.8%)								*Target 320 Tokens (↓ 88.9%)*			
PDrop CVPR'25	128	61.1	1664	82.3	69.9	55.1	95.0 (↓ 5.0)	SparseVLM ICML'25	1694	67.3	93.9 (↓ 6.1)
SparseVLM ICML'25	128	64.5	1746	85.0	68.6	56.7	97.7 (↓ 2.3)	VisionZip CVPR'25	1702	67.3	94.1 (↓ 5.9)
VisionZip CVPR'25	128	62.0	1762	83.2	68.9	56.8	96.8 (↓ 3.2)	DivPrune CVPR'25	1731	67.8	95.3 (↓ 4.7)
DivPrune CVPR'25	128	61.6	1713	87.5	68.7	55.9	96.8 (↓ 3.2)	VisionTrim ICLR'26	1795	69.6	98.3 (↓ 1.7)
ZOO-Prune CVPR'26	128	61.9	1752	87.1	68.9	57.9	97.9 (↓ 2.1)	ZOO-Prune CVPR'26	1788	67.8	96.8 (↓ 3.2)
Ours	**123** (↓ 78.6%)	**64.6**	**1795**	**86.0**	**69.7**	**56.8**	**98.8** (↓ 1.2)	**Ours**	**1796**	**73.7**	**101.2** (↑ 1.2)
Target 64 Tokens (↓ 88.9%)								*Target 160 Tokens (↓ 94.4%)*			
PDrop CVPR'25	64	33.3	1092	55.9	49.0	45.9	64.9 (↓ 35.1)	SparseVLM ICML'25	1542	67.5	89.9 (↓ 10.1)
SparseVLM ICML'25	64	60.1	1589	77.5	52.0	53.4	87.0 (↓ 13.0)	VisionZip CVPR'25	1630	68.3	92.9 (↓ 7.1)
VisionZip CVPR'25	64	60.1	1690	77.0	69.0	55.5	93.6 (↓ 6.4)	DivPrune CVPR'25	1658	68.0	93.4 (↓ 6.6)
DivPrune CVPR'25	64	60.1	1619	86.2	68.0	54.6	94.3 (↓ 5.7)	VisionTrim ICLR'26	1702	70.2	96.2 (↓ 3.8)
ZOO-Prune CVPR'26	64	60.2	1676	85.9	68.3	55.4	95.3 (↓ 4.7)	ZOO-Prune CVPR'26	1739	68.4	95.9 (↓ 4.1)
Ours	**60** (↓ 89.6%)	**62.2**	**1717**	**84.3**	**70.0**	**55.1**	**96.4** (↓ 3.6)	**Ours**	**1787**	**72.9**	**100.4** (↑ 0.4)



(a) LLM-Aligned Evidence Across Compression (b) Comparison with Existing Methods

Figure 4: **Qualitative comparison.** LAVIC better retains answer-relevant visual evidence under progressive compression than existing methods (Zhang et al., 2025b; Yang et al., 2025; Kim et al., 2026). Yellow dotted boxes are shown only for visualization and are not used during inference.

reduces the compression scale and preserves more visual capacity. We use $\alpha = 0.7$ and $\beta = 0.6$ as the default settings.

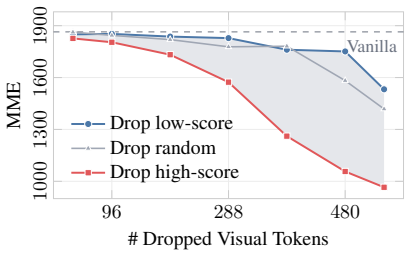
4 Experiments

4.1 Experimental Setup

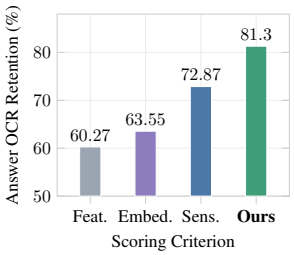
Our experiments span diverse VLM architectures and visual input regimes, including standard images, high-resolution and multi-image inputs, and videos. We evaluate LAVIC on LLaVA-v1.5 (Liu et al., 2024a), LLaVA-NeXT (Liu et al., 2024b), LLaVA-OneVision (Li et al., 2024), InternVL2.5 (Chen et al., 2024b), Mini-Gemini (Li et al., 2025), Qwen-VL-Chat (Bai et al., 2023), Qwen3.5-4B (Qwen Team, 2026), Gemma-4-E4B-it (Gemma Team, 2026), and Video-LLaVA (Lin et al., 2024). Beyond downstream performance, we analyze how LAVIC utilizes LLM-aligned visual evidence throughout compression, as well as its efficiency and robustness. Additional implementation and evaluation details are provided in Appendix A.

Criterion	MME
Random	1751
Feature magnitude	1724
Sensitivity	1776
Attention	1788
Ours	**1795**

(a) Criterion comparison

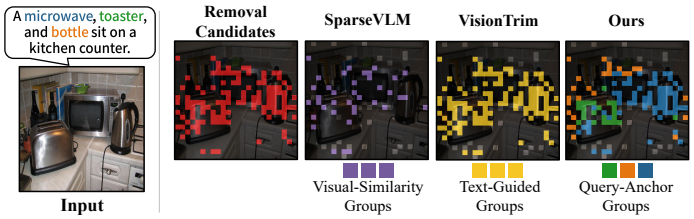


(b) High- vs. low-score removal



(c) Answer OCR retention

Figure 5: **Analysis of LLM-expressive evidence on LLaVA-v1.5-7B (Liu et al., 2024a).** Sensitivity and attention use the scoring schemes from ZOO-Prune (Kim et al., 2026) and VisionZip (Yang et al., 2025), respectively.



(a) Grouping visualization

Grouping	POPE	SQA	VQA^{Text}
Visual Similarity	83.8	69.4	53.9
Text-Guided	83.7	69.4	54.2
Ours	**84.3**	**70.1**	**55.1**

(b) Quantitative comparison

Figure 6: **Comparison of removal-candidate grouping strategies on LLaVA-v1.5-7B (Liu et al., 2024a).** Visual-similarity and text-guided grouping use the schemes from SparseVLM (Zhang et al., 2025b) and VisionTrim (Yu et al., 2026), respectively.

Table 2: **Analysis of query-anchored evidence on Qwen-VL-Chat (Bai et al., 2023).**

(a) Grouping and summarization.

Grouping	Summary	MME	FLOPs (T)	Prefill (ms)
Random	Weighted	1732	0.79	26.8
Visual Sim	Weighted	1734	0.79	30.8
Query	Mean	1687	0.69	25.9
Query	Max	1744	0.69	26.9
Query	**Weighted**	**1758**	0.76	26.5

(b) Attention.

Metric	Retained	Summary
Proportion	92%	8%
Rel. attention		
Layers 16–31	0.298 ± 0.326	
Final layer	0.795 ± 0.455	

Table 3: **Evidence-aware capacity on Qwen-VL-Chat (Bai et al., 2023).**

Allocation	MME	SEED
Token reduction ↓ 69.5%		
Fixed	1738	62.3
Ours	**1758**	**63.0**
Token reduction ↓ 89.1%		
Fixed	1336	50.9
Ours	**1369**	**51.4**

4.2 Main Results

Table 1 compares LAVIC with representative training-free baselines on LLaVA-v1.5-7B and LLaVA-NeXT-7B under comparable visual-token budgets. Across both models, LAVIC maintains stronger aggregate performance than the compared methods at every evaluated budget, with the margin becoming more pronounced as compression becomes more aggressive. Even with only 60 of 576 visual tokens on LLaVA-v1.5-7B, LAVIC retains 96.4% of the uncompressed aggregate performance, while similarly strong performance is maintained on LLaVA-NeXT under substantial token reduction. Fig. 4 further illustrates how LAVIC preserves LLM-aligned visual evidence throughout compression. Answer-relevant regions remain visible across the LLM-facing filtering and subsequent LLM compression stages, while existing methods more often discard or fragment such evidence. These qualitative results further support LLM-aligned visual evidence as an effective basis for visual token compression, particularly under aggressive compression. Broader comparisons with additional baselines and model settings are provided in Appendix B.

4.3 Analysis of LLM-Aligned Visual Evidence

LLM-Expressive Evidence. We examine whether prioritizing tokens based on the distinguishability of their projected local variations identifies visual evidence aligned with downstream LLM

Table 4: **Generalization to Video-LLaVA and InternVL2.5-8B**, covering video and high-resolution/multi-image settings. Acc. denotes the average relative performance across benchmarks.

Method	Video-LLaVA (Lin et al., 2024)								InternVL2.5-8B (Chen et al., 2024b)						
	MSVD-QA		TGIF-QA		MSRVTT-QA		FLOPs (T)	Prefill (ms)	BLINK	Muir	MMT	MMMU	Info	Acc. (%)	FLOPs (T)
	Acc	Score	Acc	Score	Acc	Score									
Upper bound, 2048 tokens (100%)									*Upper bound (100%)*						
Vanilla	75.7	3.76	54.7	2.72	64.0	3.18	14.75	440.8	55.0	51.1	62.5	54.3	72.6	100	30.34
Token reduction ↓ 66.8%									*Token reduction ↓ 51.0%*						
SparseVLM ICML'25	55.2	2.81	46.9	2.44	45.7	2.32	5.31	134.2	54.6	50.9	62.1	53.4	64.4	97.1	9.09
Ours	**55.3**	**2.82**	**47.3**	**2.45**	**45.5**	**2.31**	**5.24**	**115.2**	**55.1**	**50.5**	**62.2**	**53.6**	**64.8**	**97.3**	**8.67**
Token reduction ↓ 77.9%									*Token reduction ↓ 70.2%*						
SparseVLM ICML'25	53.0	2.71	45.0	2.34	43.7	2.23	3.85	95.1	53.5	47.8	60.7	52.6	49.9	90.7	6.09
Ours	**54.4**	**2.78**	**46.8**	**2.43**	**44.7**	**2.28**	**3.63**	**77.5**	**53.9**	**48.6**	**60.8**	**52.3**	**53.0**	**91.9**	**5.86**
Token reduction ↓ 89.1%									*Token reduction ↓ 92.7%*						
SparseVLM ICML'25	44.2	2.30	36.6	1.94	38.6	1.98	2.34	66.1	43.3	30.5	52.1	48.2	25.9	69.3	3.16
Ours	**54.5**	**2.79**	**47.6**	**2.48**	**44.1**	**2.25**	**1.91**	**43.9**	**45.6**	**36.3**	**52.9**	**48.8**	**32.7**	**74.7**	**2.46**

Table 5: **Generalization to Qwen3.5-4B and Gemma-4-E4B-it**, representing recent VLMs with different model families and multimodal designs. Acc. denotes the average relative performance over the three benchmarks of each model.

Method	Qwen3.5-4B (Qwen Team, 2026)				Gemma-4-E4B-it (Gemma Team, 2026)			
	MME	POPE	VQA^{Text}	Acc. (%)	MME	POPE	SQA	Acc. (%)
Upper bound, full tokens (100%)								
Vanilla	2341	89.5	82.3	100	1626	87.2	84.4	100
Target 128 Tokens								
VisionZip CVPR'25	2287	88.8	60.3	90.1 (↓ 9.9)	1567	86.8	84.3	98.6 (↓ 1.4)
ZOO-Prune CVPR'26	2241	89.6	53.7	87.0 (↓ 13.0)	1318	80.6	84.1	91.0 (↓ 9.0)
Ours	**2301**	**89.3**	**64.0**	**91.9** (↓ 8.1)	**1610**	**85.7**	**84.6**	**99.2** (↓ 0.8)
Target 64 Tokens								
VisionZip CVPR'25	2097	86.2	36.2	76.6 (↓ 23.4)	1328	84.6	83.7	92.6 (↓ 7.4)
ZOO-Prune CVPR'26	2067	89.1	37.5	77.8 (↓ 22.2)	1181	77.3	83.2	86.6 (↓ 13.4)
Ours	**2296**	**89.5**	**62.0**	**91.1** (↓ 8.9)	**1605**	**85.8**	**84.3**	**99.0** (↓ 1.0)

processing. As shown in Fig. 5a, alternative prioritization criteria consistently yield lower MME performance under the same compression pipeline. Fig. 5b further shows that removing high-score tokens causes substantially greater performance degradation than removing low-score or random tokens, indicating that our criterion assigns higher scores to evidence that is more important for downstream LLM processing. Fig. 5c and Fig. 1b further show that our criterion retains substantially more answer-matching OCR evidence than alternative prioritization signals. Together, these results show that LAVIC more effectively identifies and preserves LLM-aligned visual evidence than existing token-importance criteria.

Query-Anchored Evidence. We examine whether organizing removal candidates by query anchors helps preserve and utilize query-relevant evidence during compression. Fig. 6 compares different grouping strategies, showing that query-anchor grouping consistently yields better downstream performance under the same compression pipeline. Table 2a further shows that query-anchor grouping with relevance-weighted summarization performs best among the compared strategies. As shown in Table 2b, the resulting group summaries account for only 8% of the visual sequence, yet continue to receive substantial attention in subsequent LLM layers. At the final layer, their per-token attention reaches 79.5% of that of retained visual tokens. Together, these results show that LAVIC more effectively preserves and carries forward query-relevant evidence from removal candidates through query-anchor-specific summaries.

Evidence-Aware Capacity. Table 3 shows that adapting visual capacity to the spread of query-relevant evidence consistently improves performance over fixed allocation at the same token-reduction levels. This supports adapting visual capacity to how broadly query-relevant evidence is distributed rather than using a fixed allocation.

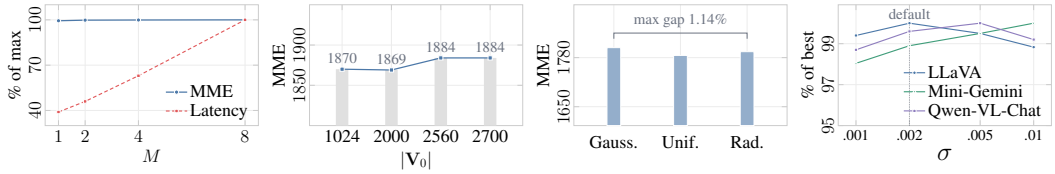
Table 6: **Runtime efficiency of LAVIC on LLaVA-v1.5-7B (Liu et al., 2024a).** LAVIC reduces computation and inference latency with modest overhead.

(a) Computation and overhead.

Metric	Vanilla	Ours
Visual tokens	576	60
TFLOPs	4.27	0.86
Prefill latency (ms)	58.70	28.95
LAVIC overhead (ms)	–	3.011
LLM-expressive evidence	–	0.326
Evidence-aware capacity	–	0.592
Query-anchored evidence	–	2.093

(b) Latency across output lengths.

N	Prefill (ms)			Decode (ms)			Total (ms)		
	Vanilla	↓66.7%	↓88.9%	Vanilla	↓66.7%	↓88.9%	Vanilla	↓66.7%	↓88.9%
16	89.8	60.8	46.6	279.1	269.0	266.4	368.9	329.8	313.0
32	90.9	62.0	45.8	581.0	556.7	551.4	671.9	618.7	597.2
64	90.9	63.2	45.5	1187.6	1136.3	1122.5	1278.5	1199.6	1167.9
128	96.2	62.4	46.0	2404.0	2306.5	2272.8	2500.3	2369.0	2318.8
256	95.3	63.8	45.9	4889.2	4675.5	4607.6	4984.5	4739.2	4653.5



(a) Local variants M. (b) Retained tokens $|\mathbf{V}_0|$. (c) Noise distribution. (d) Noise scale σ.

Figure 7: **Robustness of LLM-expressive evidence prioritization.** Performance remains stable across local variants, $|\mathbf{V}_0|$, and noise settings. Panels (a) and (b) are evaluated on Mini-Gemini (Li et al., 2025), (c) on LLaVA-v1.5-7B (Liu et al., 2024a), and (d) across multiple VLMs.

4.4 Generalization across VLM Settings

We further evaluate LAVIC across diverse VLM settings, including high-resolution and multi-image inputs (Chen et al., 2024b), video understanding (Lin et al., 2024), and recent architectures (Qwen Team, 2026; Gemma Team, 2026), to assess its generalizability beyond the main LLaVA benchmarks. As shown in Tables 4-5, LAVIC generalizes consistently across these settings, with larger gains as the visual-token budget becomes more constrained. These results show that LLM-aligned visual evidence provides an effective basis for compression across diverse VLM architectures and input regimes, particularly under aggressive compression. Additional evaluations on other VLM architectures are provided in Appendix C.

4.5 Efficiency and Robustness

Table 6a shows that LAVIC substantially reduces computation and prefill latency with modest additional overhead. Table 6b further shows that LAVIC reduces end-to-end latency across different output lengths. Fig. 7 evaluates how robust LLM-expressive evidence prioritization is to different design choices. Increasing M beyond 2 provides only marginal performance gains while increasing latency, supporting $M = 2$ as an efficient default. Performance remains stable across different values of $|\mathbf{V}_0|$, indicating that LLM-expressive evidence prioritization remains effective across different retained visual-sequence sizes. Different noise distributions yield similar performance, showing that the criterion remains effective across different ways of generating local variations. Performance also remains stable across noise scales σ and VLMs, further demonstrating its robustness across perturbation settings and model architectures.

5 Conclusion

We introduced LAVIC, a training-free visual-token compression framework that shifts the focus from identifying important tokens to preserving LLM-aligned visual evidence. LAVIC applies this principle throughout compression by prioritizing evidence expressed in the LLM-facing representation, carrying query-relevant information from removal candidates through query-anchor-specific summaries, and adapting visual capacity to evidence spread. Across diverse VLM architectures and visual input regimes, LAVIC achieves favorable performance-efficiency trade-offs, with particularly clear advantages under aggressive compression. Our analyses further support that preserving and utilizing LLM-aligned visual evidence provides an effective basis for visual token compression.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation, provide feedback on aspects of the experimental methodology and its presentation, assist with parts of the implementation, and help draft and refine parts of the manuscript. We also used these tools to help create or revise conceptual scientific figures and identify relevant literature. All AI-assisted content incorporated into this work was reviewed and verified by the authors. The authors take full responsibility for the accuracy and integrity of the submitted work.

REFERENCES

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

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

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 19–35, 2024a.

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

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1383–1392, 2024.

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 (CVPR)*, 2026.

Sanghyeok Lee, Joonmyung Choi, and Hyunwoo J. Kim. Multi-criteria token fusion with one-step-ahead attention for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15741–15750, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

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.

Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-Gemini: Mining the potential of multi-modality vision language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 292–305, 2023b.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

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

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

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. `https://llava-vl.github.io/blog/2024-01-30-llava-next/`, 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 216–233, 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2507–2521, 2022.

Minesh Mathew, Viraj Bagal, Rubén Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. InfographicVQA. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1697–1706, 2022.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 13937–13949, 2021.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22857–22867, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8317–8326, 2019.

Zicong Tang, Ziyang Ma, Suqing Wang, Zuchao Li, Lefei Zhang, Hai Zhao, Yun Li, and Qianren Wang. CoViPAL: Layer-wise contextualized visual token pruning for large vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 20701–20714, 2025.

Fei Wang, Xingyu Fu, James Y. Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, et al. MuirBench: A comprehensive benchmark for robust multi-image understanding. *arXiv preprint arXiv:2406.09411*, 2024.

Yahong Wang, Juncheng Wu, Zhangkai Ni, Chengmei Yang, Yihang Liu, Longzhen Yang, Yuyin Zhou, Ying Wen, and Lianghua He. EntropyPrune: Matrix entropy guided visual token pruning for multimodal large language models. *arXiv preprint arXiv:2602.17196*, 2026.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for important tokens in multimodal language models: Duplication matters more. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9961–9980, 2025.

Zhenkai Wu, Xiaowen Ma, Zhenliang Ni, Dengming Zhang, Han Shu, Xin Jiang, and Xinghao Chen. VLM-Pruner: Buffering for spatial sparsity in an efficient VLM centrifugal token pruning paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14593–14603, 2025.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802, 2025.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. ATP-LLaVA: Adaptive token pruning for large vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24972–24982, 2025.

Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10809–10818, 2022.

Hanxun Yu, Wentong Li, Xuan Qu, Song Wang, Junbo Chen, and Jianke Zhu. VisionTrim: Unified vision token compression for training-free MLLM acceleration. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. MM-Vet: Evaluating large multimodal models for integrated capabilities. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

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 (CVPR)*, pp. 9556–9567, 2024.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in MLLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 74840–74857, 2025b.

A Additional Experimental Details

Evaluation Protocols. We follow the standard evaluation protocol for each benchmark. For image understanding across standard, high-resolution, and multi-image settings, we evaluate on MME (Fu et al., 2025), MMMU (Yue et al., 2024), MMBench (Liu et al., 2024c), POPE (Li et al., 2023b), ScienceQA (Lu et al., 2022), MuirBench (Wang et al., 2024), InfoVQA (Mathew et al., 2022), VQA^{Text} (Singh et al., 2019), MM-Vet (Yu et al., 2024), and SEED-Bench (Li et al., 2023a). For video question answering, we follow the Video-LLaVA (Lin et al., 2024) evaluation protocol and use Llama-3-8B-Instruct (Grattafiori et al., 2024) for answer scoring. We re-evaluate baselines under matched settings when possible and otherwise report published results under comparable protocols.

Backbones. We evaluate LAVIC on LLaVA-v1.5 (Liu et al., 2024a), LLaVA-NeXT (Liu et al., 2024b), LLaVA-OneVision (Li et al., 2024), InternVL2.5 (Chen et al., 2024b), Mini-Gemini (Li et al., 2025), Qwen-VL-Chat (Bai et al., 2023), Qwen3.5-4B (Qwen Team, 2026), Gemma-4-E4B-it (Gemma Team, 2026), and Video-LLaVA (Lin et al., 2024). This broad set of VLMs allows us to assess generalization across architectures, visual input regimes, and modalities. The backbone used for each analysis is specified in the corresponding table or figure.

Efficiency Measurement. Latency is measured on a single NVIDIA RTX 4090 with batch size 1, and FLOPs are reported under the same input settings. Prefill latency excludes decoding, whereas end-to-end latency includes autoregressive generation at the specified output length.

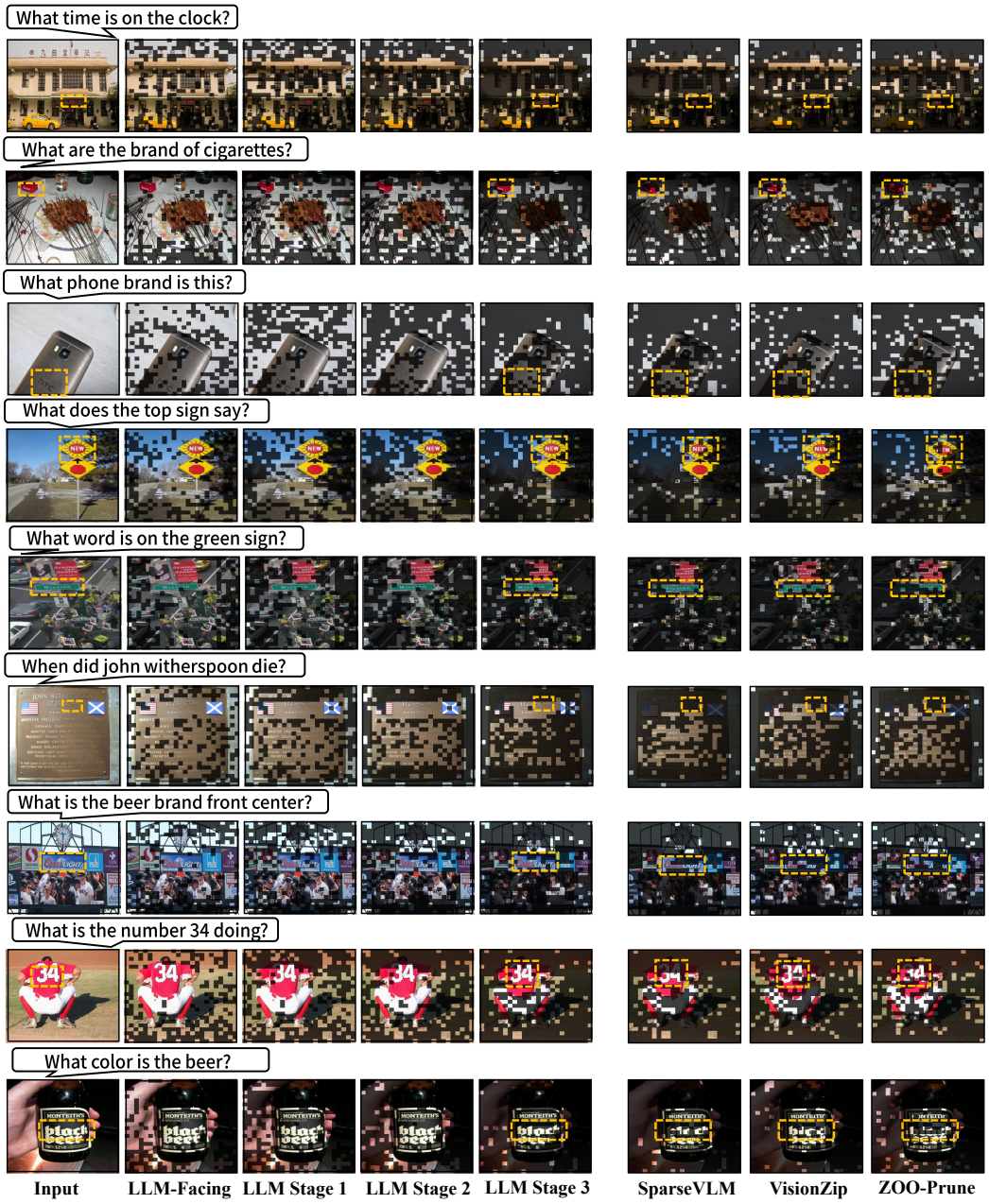
Table 7: **Extended comparison on LLaVA-v1.5-7B (Liu et al., 2024a) at comparable visual-token budgets.** T-free denotes training-free methods, Tokens the average retained visual-token count, and Acc. the mean normalized score across five benchmarks. A dash denotes an unreported metric; † denotes Acc. averaged over the four reported benchmarks.

Method	T-free	Tokens	MMB	MME	POPE	SQA	VQA^{Text}	Acc. (%)
Upper bound, 576 tokens (100%)								
Vanilla (Liu et al., 2024a)	–	576	64.6 100%	1864 100%	85.9 100%	69.5 100%	58.3 100%	100
Target 192 Tokens (↓ 66.7%)								
ToMe (Bolya et al., 2023)	✓	192	60.5 93.7%	1563 83.9%	72.4 84.3%	65.2 93.8%	52.1 89.4%	89.0 (↓ 11.0)
FastV (Chen et al., 2024a)	✓	192	61.0 94.4%	1605 86.1%	64.8 75.4%	69.1 99.4%	49.1 84.2%	87.9 (↓ 12.1)
PDrop (Xing et al., 2025)	✓	192	63.2 97.8%	1766 94.7%	82.3 95.8%	70.2 101.0%	56.1 96.2%	97.1 (↓ 2.9)
SparseVLM (Zhang et al., 2025b)	✓	192	64.1 99.2%	1787 95.9%	85.3 99.3%	68.7 98.8%	57.8 99.1%	98.5 (↓ 1.5)
DART (Wen et al., 2025)	✓	192	63.6 98.5%	1855 99.5%	82.9 96.5%	69.1 99.4%	57.5 98.6%	98.5 (↓ 1.5)
VisionZip (Yang et al., 2025)	✓	192	63.0 97.5%	1783 95.7%	85.3 99.3%	68.9 99.1%	57.3 98.3%	98.0 (↓ 2.0)
DivPrune (Alvar et al., 2025)	✓	192	62.0 96.0%	1771 95.0%	87.7 102.1%	68.7 98.8%	56.6 97.1%	97.8 (↓ 2.2)
CDPruner (Zhang et al., 2025a)	✓	192	63.2 97.8%	1786 95.8%	–	68.9 99.1%	57.3 98.3%	97.8† (↓ 2.2)
PruMerge (Shang et al., 2025)	✓	192	56.3 87.2%	1515 81.3%	–	63.6 91.5%	50.1 85.9%	86.5† (↓ 13.5)
EntropyPrune (Wang et al., 2026)	✓	192	64.4 99.7%	1844 98.9%	–	69.1 99.4%	57.6 98.8%	99.2† (↓ 0.8)
ZOO-Prune (Kim et al., 2026)	✓	192	62.9 97.4%	1782 95.6%	87.2 101.5%	69.2 99.6%	57.3 98.3%	98.5 (↓ 1.5)
VLM-Pruner (Wu et al., 2026)	✓	192	63.5 98.3%	1788 95.9%	86.0 100.1%	70.0 100.7%	58.0 99.5%	98.9 (↓ 1.1)
Ours (↓ 68.9%)	✓	179	**64.7** 100.2%	**1798** 96.5%	**85.9** 100.0%	**69.8** 100.4%	**57.5** 98.6%	**99.1** (↓ 0.9)
Target 128 Tokens (↓ 77.8%)								
ToMe (Bolya et al., 2023)	✓	128	53.3 82.5%	1490 79.9%	62.8 73.1%	59.6 85.8%	49.1 84.2%	81.1 (↓ 18.9)
FastV (Chen et al., 2024a)	✓	128	56.1 86.8%	1490 79.9%	53.4 62.2%	68.6 98.7%	50.5 86.6%	82.9 (↓ 17.1)
PDrop (Xing et al., 2025)	✓	128	61.1 94.6%	1664 89.3%	82.3 95.8%	69.9 100.6%	55.1 94.5%	95.0 (↓ 5.0)
SparseVLM (Zhang et al., 2025b)	✓	128	64.5 99.8%	1746 93.7%	85.0 99.0%	68.6 98.7%	56.7 97.3%	97.7 (↓ 2.3)
DART (Wen et al., 2025)	✓	128	63.1 97.7%	1826 98.0%	80.7 93.9%	69.0 99.3%	56.4 96.7%	97.1 (↓ 2.9)
VisionZip (Yang et al., 2025)	✓	128	62.0 96.0%	1762 94.5%	83.2 96.9%	68.9 99.1%	56.8 97.4%	96.8 (↓ 3.2)
DivPrune (Alvar et al., 2025)	✓	128	61.6 95.4%	1713 91.9%	87.5 101.9%	68.7 98.8%	55.9 95.9%	96.8 (↓ 3.2)
CDPruner (Zhang et al., 2025a)	✓	128	63.0 97.5%	1775 95.2%	–	69.0 99.3%	56.2 96.4%	97.1† (↓ 2.9)
PruMerge (Shang et al., 2025)	✓	128	54.6 84.5%	1518 81.4%	–	63.8 91.8%	50.1 85.9%	85.9† (↓ 14.1)
ATP-LLaVA (Ye et al., 2025)	✗	144	66.0 102.2%	1474 79.1%	84.2 98.0%	69.1 99.4%	–	94.7† (↓ 5.3)
EntropyPrune (Wang et al., 2026)	✓	128	63.5 98.3%	1780 95.5%	–	69.4 99.9%	56.3 96.6%	97.6† (↓ 2.4)
ZOO-Prune (Kim et al., 2026)	✓	128	61.9 95.8%	1752 94.0%	87.1 101.4%	68.9 99.1%	57.9 99.3%	97.9 (↓ 2.1)
VLM-Pruner (Wu et al., 2026)	✓	128	62.5 96.7%	1768 94.8%	85.4 99.4%	69.3 99.7%	57.3 98.3%	97.8 (↓ 2.2)
Ours (↓ 78.6%)	✓	123	**64.6** 100.0%	**1795** 96.3%	**86.0** 100.1%	**69.7** 100.3%	**56.8** 97.4%	**98.8** (↓ 1.2)
Target 64 Tokens (↓ 88.9%)								
ToMe (Bolya et al., 2023)	✓	64	43.7 67.6%	1138 61.1%	52.5 61.1%	50.6 72.8%	45.3 77.7%	68.1 (↓ 31.9)
FastV (Chen et al., 2024a)	✓	64	47.2 73.1%	1255 67.3%	38.2 44.5%	68.7 98.8%	47.8 82.0%	73.1 (↓ 26.9)
PDrop (Xing et al., 2025)	✓	64	33.3 51.5%	1092 58.6%	55.9 65.1%	49.0 70.5%	45.9 78.7%	64.9 (↓ 35.1)
SparseVLM (Zhang et al., 2025b)	✓	64	60.1 93.0%	1589 85.2%	77.5 90.2%	52.0 74.8%	53.4 91.6%	87.0 (↓ 13.0)
DART (Wen et al., 2025)	✓	64	60.6 93.8%	1751 93.9%	74.6 86.8%	68.7 98.8%	54.3 93.1%	93.3 (↓ 6.7)
VisionZip (Yang et al., 2025)	✓	64	60.1 93.0%	1690 90.7%	77.0 89.6%	69.0 99.3%	55.5 95.2%	93.6 (↓ 6.4)
DivPrune (Alvar et al., 2025)	✓	64	60.1 93.0%	1619 86.9%	86.2 100.3%	68.0 97.8%	54.6 93.7%	94.3 (↓ 5.7)
ATP-LLaVA (Ye et al., 2025)	✗	88	64.7 100.2%	1402 75.2%	82.6 96.2%	67.2 96.7%	–	92.1† (↓ 7.9)
ZOO-Prune (Kim et al., 2026)	✓	64	60.2 93.2%	1676 89.9%	85.9 100.0%	68.3 98.3%	55.4 95.0%	95.3 (↓ 4.7)
VLM-Pruner (Wu et al., 2026)	✓	64	61.3 94.9%	1752 94.0%	82.2 95.7%	68.4 98.4%	56.1 96.2%	95.8 (↓ 4.2)
Ours (↓ 89.6%)	✓	60	**62.2** 96.3%	**1717** 92.1%	**84.3** 98.1%	**70.0** 100.7%	**55.1** 94.5%	**96.4** (↓ 3.6)

B Extended Main Results

B.1 Full Comparison under Visual-Token Budgets

Table 7 extends the main comparison on LLaVA-v1.5-7B (Liu et al., 2024a) to a broader set of visual-token compression methods under comparable visual-token budgets. LAVIC performs consistently well across visual-token budgets, with clearer advantages under aggressive compression. These results further support LLM-aligned visual evidence as an effective basis for compression under tight token budgets.



(a) LLM-Aligned Evidence Across Compression (b) Comparison with Existing Methods

Figure 8: **Extended qualitative comparison on LLaVA-v1.5-7B (Liu et al., 2024a).** LAVIC retains answer-relevant visual evidence across progressive compression more consistently than existing methods (Zhang et al., 2025b; Yang et al., 2025; Kim et al., 2026). Yellow dotted boxes are for visualization only and are not used during inference.

B.2 Extended Qualitative Results

Fig. 8 provides additional qualitative comparisons across diverse question types, including text reading, object and brand recognition, action, and attribute questions. LAVIC consistently preserves answer-relevant visual evidence across progressive compression, whereas existing methods can discard or obscure such evidence under tight token budgets.

Table 8: **Results on LLaVA-OneVision-7B (Li et al., 2024).**

Method	Tokens	MME	MMStar	SEED	Acc. (%)
Upper bound, 729 tokens (100%)					
Vanilla (Li et al., 2024)	729	1605	61.7	76.6	100
Token reduction ↓ 50.0%					
FastV (Chen et al., 2024a)	365	1586	57.2	75.0	96.5 (↓ 3.5)
SparseVLM (Zhang et al., 2025b)	365	1560	54.7	74.4	94.3 (↓ 5.7)
PDrop (Xing et al., 2025)	365	1533	59.0	75.8	96.7 (↓ 3.3)
CoViPAL (Tang et al., 2025)	365	1613	59.1	75.5	98.3 (↓ 1.7)
Ours	**362**	**1614**	**56.3**	**75.6**	**96.8** (↓ 3.2)
Token reduction ↓ 75.0%					
FastV (Chen et al., 2024a)	182	1523	47.2	65.9	85.8 (↓ 14.2)
SparseVLM (Zhang et al., 2025b)	182	1416	45.4	67.4	83.3 (↓ 16.7)
PDrop (Xing et al., 2025)	182	1470	49.8	67.5	86.8 (↓ 13.2)
CoViPAL (Tang et al., 2025)	182	1559	54.3	73.2	93.6 (↓ 6.4)
Ours	**177**	**1635**	**54.4**	**73.9**	**95.5** (↓ 4.5)

Table 9: **Results on Mini-Gemini (Li et al., 2025).**

Method	Tokens	MME	MMB	MMVet	VQA^{Text}	Acc. (%)	FLOPs (T)
Vanilla	2880	1879	64.9	40.3	68.4	100.0	22.1
Ours	887	1884	65.6	36.9	66.1	97.4	7.3
	600	1874	65.6	29.7	63.4	91.8	5.1
	283	1857	64.4	28.7	57.9	88.5	2.7

Table 10: **Results on Qwen-VL-Chat (Bai et al., 2023).**

Reduction	Method	Tokens	MME	SEED	Acc. (%)	FLOPs (T)
Full	Vanilla	256	1820	64.6	100	2.04
↓69.5%	SparseVLM	79	1709	60.6	93.8	0.88
	Ours	76	1758	63.0	97.0	0.86
↓89.1%	SparseVLM	30	1334	49.7	75.2	0.56
	Ours	26	1369	51.4	77.4	0.53

C Additional Results Across VLMs

We further evaluate LAVIC across VLMs with different visual representations and visual-token regimes.

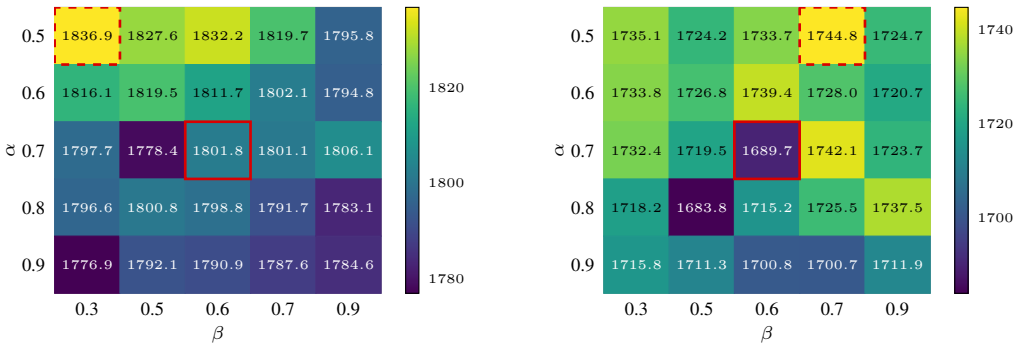
LLaVA-OneVision. LLaVA-OneVision (Li et al., 2024) supports varying image resolutions and aspect ratios through its AnyRes design, providing a different visual representation from LLaVA-v1.5. As shown in Table 8, LAVIC remains competitive at moderate compression and achieves the highest aggregate performance at the tighter token budget, showing clearer advantages as compression becomes more aggressive.

Mini-Gemini. Mini-Gemini (Li et al., 2025) employs dual vision encoders for high-resolution visual representation and processes a long visual sequence of 2,880 tokens. As shown in Table 9, LAVIC retains 97.4% of the uncompressed aggregate performance with 887 visual tokens while reducing FLOPs from 22.1 to 7.3 T.

Qwen-VL-Chat. Qwen-VL-Chat (Bai et al., 2023) provides a complementary compact-token setting, compressing visual features into a fixed sequence of 256 tokens before LLM processing. As shown in Table 10, LAVIC maintains strong performance at both compression levels, showing that LLM-aligned visual compression remains effective under an already compact visual-token regime.

Table 11: **Per-benchmark TFLOPs on LLaVA-v1.5-7B (Liu et al., 2024a) under comparable visual-token budgets.** LAVIC consistently reduces computation across benchmarks.

Method	Tokens	MME	SQA	MMB	VQA^{Text}	POPE	MMVet	Avg.
Vanilla	576	4.27	4.72	4.62	4.61	4.24	4.20	4.44
SparseVLM (Zhang et al., 2025b)	192	1.79	2.23	2.14	2.12	1.76	1.73	1.96
Ours	**179**	**1.70**	**2.14**	**2.05**	**2.06**	**1.68**	**1.64**	**1.88**
SparseVLM (Zhang et al., 2025b)	128	1.39	1.82	1.96	1.72	1.36	1.54	1.63
Ours	**123**	**1.31**	**1.76**	**1.67**	**1.66**	**1.28**	**1.25**	**1.49**
SparseVLM (Zhang et al., 2025b)	64	0.97	1.41	1.32	1.30	0.94	0.91	1.14
Ours	**62**	**0.86**	**1.32**	**1.22**	**1.23**	**0.83**	**0.80**	**1.04**



(a) Compression ↓66.8%

(b) Compression ↓89.6%

Figure 9: **Sensitivity of Evidence-Aware Capacity to α and β.** MME performance on LLaVA-v1.5-7B (Liu et al., 2024a) under two compression levels. Solid and dashed boxes indicate the default $(0.7, 0.6)$ and the best-performing settings, respectively.

D Additional Efficiency Analysis

Table 11 reports per-benchmark computation on LLaVA-v1.5-7B (Liu et al., 2024a) under comparable visual-token budgets. LAVIC consistently requires less computation across benchmarks and compression levels, showing that its efficiency gains are not limited to a particular task.

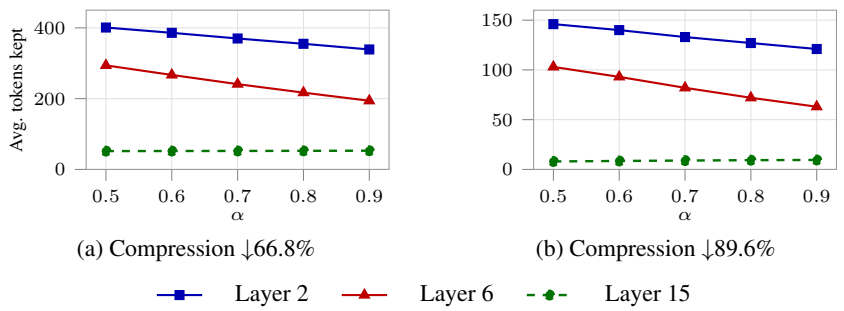
E Additional Analysis of LAVIC

E.1 Evidence-Aware Capacity

We further analyze the hyperparameter sensitivity and layer-wise behavior of Evidence-Aware Capacity. In Eq. 5, α controls the base compression strength, while β determines how strongly the capacity adapts to evidence spread.

Sensitivity to α and β. Fig. 9 reports MME performance across different α and β values under two compression levels on LLaVA-v1.5-7B (Liu et al., 2024a). Performance generally decreases as α increases, while the preferred β varies with α and the compression level. We use a single fixed setting of $(\alpha, \beta) = (0.7, 0.6)$ throughout all experiments without compression-specific tuning.

Layer-Wise Capacity Allocation. Fig. 10 examines how α affects layer-wise visual capacity. Across both compression levels, increasing α reduces token retention mainly at early and middle layers, while the retained-token count at the last pruning layer remains nearly unchanged. This indicates that α mainly controls how aggressively visual tokens are compressed at intermediate LLM stages, rather than substantially changing the final retained-token count.



(a) Compression ↓66.8% (b) Compression ↓89.6%

Figure 10: **Layer-wise effect of α on Evidence-Aware Capacity.** Average retained visual-token counts at different pruning layers on LLaVA-v1.5-7B (Liu et al., 2024a) under two compression levels.

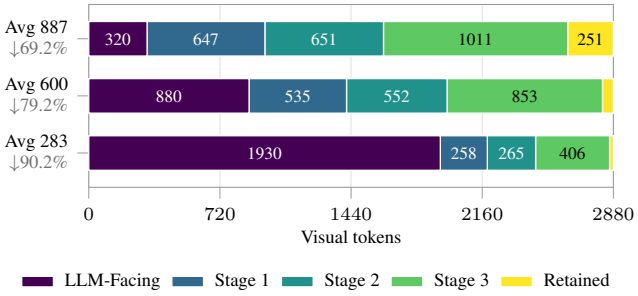


Figure 11: **Stage-wise visual-token reduction on Mini-Gemini (Li et al., 2025).** LAVIC progressively reduces visual tokens across LLM-facing filtering and subsequent LLM compression stages under three compression levels.

Table 12: **Stability across random seeds on LLaVA-v1.5-7B (Liu et al., 2024a).** Mean±Std and performance range over eight random seeds.

Bench.	Mean±Std	Range
VQA^{Text}	56.09 ± 0.26	0.65
POPE	86.02 ± 0.19	0.53
SQA	69.02 ± 0.21	0.60

E.2 Compression Dynamics

Fig. 11 visualizes the compression trajectory of LAVIC on Mini-Gemini (Li et al., 2025) under three compression levels. Rather than relying on a single reduction step, LAVIC progressively compresses the visual sequence using LLM-expressive evidence before LLM processing and query-relevant evidence within the LLM.

E.3 Stability Across Random Seeds

As shown in Table 12, LAVIC exhibits stable performance across eight random seeds on LLaVA-v1.5-7B (Liu et al., 2024a). The standard deviations remain below 0.3, and the performance ranges remain below 0.7 across all three benchmarks.

E.4 Failure Analysis

LAVIC can still struggle when answering requires fine-grained discrimination among closely related visual evidence. On VQA^{Text}, among 148 failure cases with identifiable answer-related OCR patches, 140 still retain the relevant answer evidence. However, in 67 of these 140 cases, the answer-related region is summarized together with a median of 10 additional non-answer OCR regions. This can make the target answer less distinguishable from nearby competing text. These cases suggest that preserving finer distinctions among query-relevant evidence during summarization could further improve performance on text-rich tasks.