Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

UniPrune: Unified Progressive Visual Token Pruning with Information-Aware Budget Allocation for Efficient LLaVA-Style Vision Language Models

Jianhua Cui
China Tobacco Henan Industrial Co., Ltd
Meizhou Ding
dingmeizhou999@163.com

Key Laboratory for Tobacco Processing Morphology Research
Ziru Niu
China Tobacco Henan Industrial Co., Ltd
Wei Wang
China Tobacco Henan Industrial Co., Ltd
Lifeng Han
Key Laboratory for Tobacco Processing Morphology Research
Peng Zhao
China Tobacco Henan Industrial Co., Ltd

Research Article

Keywords: Vision language models, Visual token pruning, Multi-stage compression, Budget allocation, Information continuity, Inference efficiency, Training-free

Posted Date: August 31st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9555450/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

UniPrune: Unified Progressive Visual Token Pruning with Information-Aware Budget Allocation for Efficient LLaVA-Style Vision Language Models

Jianhua Cui[2], Meizhou Ding[1,2*], Ziru Niu[2], Wei Wang[2*], Lifeng Han[1,2], Peng Zhao[2]

[1]Key Laboratory for Tobacco Processing Morphology Research, China.
[2]China Tobacco Henan Industrial Co., Ltd, China.

*Corresponding author(s). E-mail(s): dingmeizhou999@163.com; a1591552064@163.com;
Contributing authors: brownaycjh@gmail.com; niuzr@hatic.com; xc_hanlf@hatic.com; 15538302070@163.com;

Abstract

LLaVA-style vision language models (VLMs) suffer from severe computational inefficiency caused by excessive visual tokens. Existing compression methods either prune tokens at the vision encoder output (*encoder-stage*) or drop tokens progressively within the LLM (*LLM-stage*), but these single-stage approaches face complementary limitations: encoder-stage methods lack text-guided context and degrade sharply at extreme compression ratios, while LLM-stage methods cannot save computation in early transformer layers where all tokens are still processed. In this paper, we present UniPrune, a unified multi-stage visual token pruning framework that synergistically combines semantic importance-diversity pruning at the encoder stage with pyramid-shaped progressive dropping within the LLM, while introducing two additional mechanisms. First, we propose **Information-Aware Budget Allocation (IABA)**, an empirically motivated strategy that allocates the token budget between the two stages based on image information density, suggesting that the preferred intermediate token count follows a near-geometric-mean rule between the initial and final token counts. Second, we introduce **Cross-Stage Information Continuity (CSIC)**, which augments encoder-stage token selection with an LLM-friendliness score derived from attention entropy in CLIP-ViT-L/14 layer 22, ensuring that the tokens

forwarded to the LLM form a high-quality foundation for subsequent LLM-stage dropping. Extensive experiments on LLaVA-1.5 and LLaVA-NeXT, two representative LLaVA-style VLM families, across five benchmarks demonstrate that UniPrune consistently outperforms the compared single-stage baselines. Notably, at extreme compression ratios (<5% token retention), UniPrune surpasses the best single-stage baseline by a clear margin while reducing total inference FLOPs and peak GPU memory.

Keywords: Vision language models, Visual token pruning, Multi-stage compression, Budget allocation, Information continuity, Inference efficiency, Training-free

1 Introduction

LLaVA-style vision language models (VLMs), exemplified by LLaVA-1.5 and LLaVA-NeXT [1–3], have achieved strong performance across a broad spectrum of multimodal tasks, including visual question answering, document understanding, and video reasoning. These models typically encode an input image into a sequence of visual tokens using a Vision Transformer (ViT) encoder, which are then processed jointly with text tokens by a large language model (LLM). However, this paradigm introduces a critical efficiency bottleneck: contemporary LLaVA-style models such as LLaVA-1.5 [1] and LLaVA-NeXT [2] generate 576 and up to 2880 visual tokens per image, respectively, despite empirical evidence [4] showing that approximately 70% of these tokens carry negligible semantic information.

The computational cost of processing visual tokens scales quadratically with sequence length in Transformer self-attention, making visual token redundancy a first-order efficiency concern, especially in high-resolution image and video settings. This has motivated a rich body of research on *visual token compression*, which broadly falls into two paradigms based on *where* compression occurs in the VLM processing pipeline:

Encoder-stage compression methods [5–9] prune tokens at the output of the vision encoder, before they are passed to the LLM. These methods exploit purely visual features, such as CLS attention scores or pairwise token similarity to identify and remove redundant tokens. By compressing early, they reduce both the projector and all LLM layers' computational burden. However, because compression decisions are made without any text-guided context, these methods cannot identify which tokens are task-relevant, and suffer severe performance degradation at extreme compression ratios (e.g., retaining fewer than 5% of tokens).

LLM-stage compression methods [10–13] instead drop tokens within the LLM's transformer layers, leveraging the cross-modal attention between text and visual tokens to guide pruning. PyramidDrop [11], for instance, empirically demonstrates that visual token redundancy *monotonically increases* with LLM layer depth, motivating a pyramid-shaped progressive dropping schedule across multiple stages. While these methods benefit from rich multimodal interaction signals, they are inherently

limited: all visual tokens must be fully processed in the LLM's early layers before any dropping occurs, yielding no savings for shallow-layer computation.

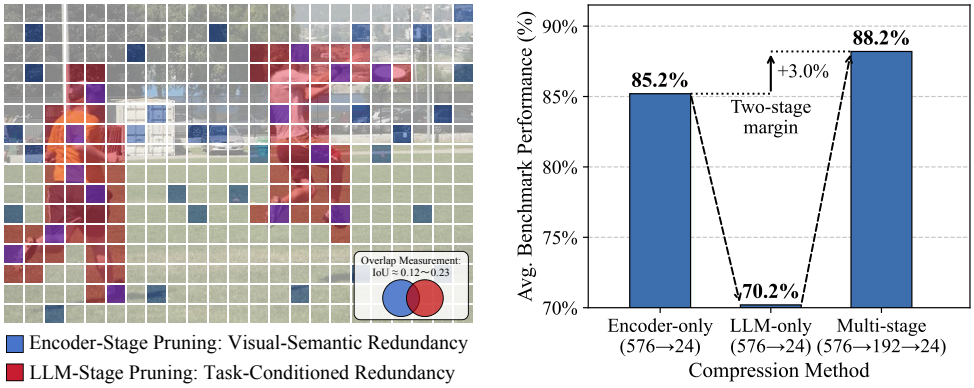


Fig. 1 Motivation for multi-stage compression. (**a**) Encoder-stage methods remove globally redundant tokens based on visual appearance alone, while LLM-stage methods target task-conditioned redundancy. The two types exhibit low overlap (IoU $\approx$ 0.12 to 0.23 on POPE [14]), confirming their complementarity. (**b**) Combining PruneSID (576→192) with PyramidDrop (192→24) substantially outperforms either single-stage approach at the same final token count of 24.

The core limitation of both paradigms is their single-stage nature, which prevents them from capturing the two fundamentally *complementary* types of visual token redundancy: (1) *visual-semantic redundancy*, globally redundant tokens based on visual appearance alone (e.g., repetitive background textures), best identified by encoder-stage methods; and (2) *task-conditioned redundancy*, tokens whose relevance diminishes as the LLM builds deeper cross-modal representations, best identified by LLM-stage methods.

We provide an empirical demonstration of this complementarity in Fig. 1: when using PruneSID [5] alone to compress from 576 tokens to 24 (4.2% retention), average benchmark performance on LLaVA-1.5 drops significantly. Using PyramidDrop [11] alone at the same budget yields a comparable level of degradation. However, combining PruneSID (576→192) with PyramidDrop (192→24) achieves a substantially higher performance level despite maintaining *identical* final token counts. This motivating observation raises an important question that has not been addressed by any prior work: **Given a fixed final token budget, how should the compression ratio be optimally allocated between the encoder stage and the LLM stage?** Naively splitting the budget is suboptimal; the allocation must account for the information density of the input image and the downstream token quality requirements for effective LLM-stage dropping.

To address these challenges, we propose UniPrune, a unified multi-stage visual token pruning framework with the following contributions:

- **Two-stage synergistic compression pipeline.** UniPrune applies PSCA-based semantic grouping [5] with intra-group NMS at the encoder stage, followed by

pyramid-shaped similarity-based dropping [11] within the LLM, systematically exploiting both types of visual redundancy in a unified framework.

- **Information-Aware Budget Allocation (IABA).** We propose an empirically motivated, image-adaptive strategy that determines the intermediate token count N_1 (the encoder-stage output budget) for a given final token count N_2. IABA computes a global information density score from the token similarity distribution and allocates proportionally more tokens to complex images, suggesting a near-geometric-mean rule: $N_1^* \approx \beta\sqrt{N \cdot N_2}$ where $\beta \in [1.3, 1.8]$.
- **Cross-Stage Information Continuity (CSIC).** We observe that *which* tokens survive the encoder stage significantly impacts the quality of LLM-stage dropping. CSIC augments encoder-stage token scoring with an LLM-friendliness score derived from attention entropy in CLIP-ViT-L/14 layer 22, ensuring that retained tokens form a more diverse and structurally balanced input for the LLM.
- **Comprehensive empirical analysis.** We provide extensive experiments on LLaVA-1.5 and LLaVA-NeXT across five benchmarks, with thorough ablations, efficiency measurements, and qualitative analysis that illuminate why multi-stage compression outperforms single-stage approaches.

2 Related Work

2.1 Encoder-Stage Visual Token Compression

A large family of token compression methods operates at the vision encoder output, prior to LLM processing. LLaVA-PruMerge [7] selects tokens with high CLS attention and merges similar ones, improving information density. VisionZip [8] identifies dominant tokens via attention and merges contextually similar ones. HiRED [9] applies a hierarchical budget allocation based on spatial image regions. FasterVLM [6] leverages efficient vision encoding strategies for accelerated processing. DivPrune [15] explicitly optimizes for diversity in the retained token set. PACT [16] combines pruning with clustering-based strategies, grouping similar tokens and selecting cluster representatives. More recently, PruneSID [5] proposes a two-stage pipeline combining Principal Semantic Component Analysis (PSCA) for semantic grouping and intra-group Non-Maximum Suppression (NMS) for redundancy removal, achieving state-of-the-art performance at extreme compression ratios.

All encoder-stage methods share a fundamental limitation: compression decisions are agnostic to the downstream text query, resulting in irreversible information loss that cannot be recovered by the LLM. This limitation becomes increasingly severe at extreme compression ratios where the margin for error is minimal.

2.2 LLM-Stage Visual Token Compression

FastV [10] observes that visual tokens become significantly less attended after the second LLM layer and proposes to drop 50% of visual tokens based on attention scores. SparseVLM [12] aggregates text-to-visual attention across layers to identify task-relevant tokens. MustDrop [13] progressively drops tokens across multiple LLM layers. DART [17] computes pairwise similarities among tokens and prunes highly

similar ones, arguing that duplication drives the majority of redundancy. PyramidDrop [11] is the most closely related LLM-stage method: it empirically shows that visual redundancy increases monotonically with layer depth, and proposes a pyramid-shaped dropping schedule that distributes compression across multiple LLM stages, resulting in 40% training speedup and 55% inference FLOPs reduction on LLaVA-NeXT with negligible performance loss.

LLM-stage methods benefit from cross-modal context but face an inherent efficiency limitation: since no tokens are dropped until after the early LLM layers complete processing, they cannot reduce computation in those layers. For models with thousands of visual tokens, this means substantial computation is wasted in the initial LLM layers processing tokens that will eventually be dropped.

2.3 Multi-Stage and Cascaded Compression

Despite the clear complementarity between encoder-stage and LLM-stage approaches, the systematic study of their combination remains largely unexplored. Libra-Merging [18] considers importance and redundancy trade-offs but operates within a single paradigm. CLS-attention-based approaches [19, 20] exploit visual cues from the CLS token for effective early-stage pruning but do not address LLM-stage redundancy. Zhang et al. [21] further extend this line by maximizing conditional diversity during token pruning. GreedyPrune [22] and Balanced Token Pruning [23] attempt to address local optimization problems in greedy selection but remain confined to a single compression stage.

3 Method

3.1 Preliminaries and Problem Formulation

VLM Pipeline. A standard VLM consists of a vision encoder $\mathcal{E}$ (typically ViT-based), a projector $\mathcal{P}$, and a large language model $\mathcal{L}$ with L transformer layers. Given an input image I, the encoder produces N visual tokens $\mathbf{X} = \mathcal{E}(I) \in \mathbb{R}^{N \times D}$, where D is the embedding dimension. After projection, these tokens are concatenated with M text tokens and fed into $\mathcal{L}$.

Problem. Given a final target token count $N_2 \ll N$, our goal is to compress visual tokens from N to N_2 with minimal performance degradation, while reducing the total inference FLOPs across both the projector and all LLM layers. Formally, we seek to find an intermediate token count N_1 ($N_2 < N_1 < N$), a Stage-1 pruning function $f_1 : \mathbb{R}^{N \times D} \to \mathbb{R}^{N_1 \times D}$, and a Stage-2 dropping schedule $\{f_{2,s}\}_{s=1}^{S}$ such that combined performance is maximized.

Computational Cost Analysis. The FLOPs of a single transformer layer with sequence length n scale as $O(n^2 d + nd^2)$. With Stage-1 compression to N_1 and Stage-2 progressive reduction from N_1 to N_2, the total FLOPs are:

$$C_{\text{UniPrune}} \approx C_{\text{enc}} + N_1^2 \cdot k_1 \cdot d + \sum_{s=1}^{S} n_s^2 \cdot (k_{s+1} - k_s) \cdot d \tag{1}$$

where k_s is the LLM layer index at which Stage-2's s-th drop occurs and n_s is the remaining token count after the s-th drop. This is substantially lower than encoder-only compression ($N_2^2 \cdot L \cdot d$, which saves maximally but incurs accuracy loss) or LLM-only compression ($N^2 \cdot k_1 \cdot d + \cdots$, which wastes full early-layer cost).

3.2 Stage 1: Semantic Importance-Diversity Pruning (SID-Prune)

We adopt and extend the PruneSID framework [5] for encoder-stage pruning.

Principal Semantic Component Analysis (PSCA). Given visual tokens $\mathbf{X} \in \mathbb{R}^{N \times D}$, we apply Sigmoid normalization and mean-centering:

$$\mathbf{X}_{\text{ctr}} = \sigma(\mathbf{X}) - \frac{1}{N} \sum_{i=1}^{N} \sigma(\mathbf{x}_i) \tag{2}$$

We compute the truncated SVD of $\mathbf{X}_{\text{ctr}}^{\top}$, obtaining the top-$K$ right singular vectors $\mathbf{V} \in \mathbb{R}^{N \times K}$. Each token i is assigned to group $g_i = \arg\max_k |V_{ik}|$, partitioning the N tokens into K semantically coherent groups $\{G_1, \ldots, G_K\}$.

Intra-Group Non-Maximum Suppression (NMS). Within each group G_k, tokens are ranked by importance score $\hat{a}_i$ (augmented with CSIC, as described in Sect. 3.5). NMS iteratively retains the top-scored token and suppresses tokens with cosine similarity above threshold τ, until the group's quota $n_k = \lfloor N_1 \cdot |G_k|/N \rfloor$ is met. The adaptive threshold is set as $\tau = \lambda \cdot \rho$, where ρ is the global diversity / information-density score defined in Eq. 6 and $\lambda = N_1/32$. This ensures semantic coverage while eliminating within-group redundancy, with automatic adaptation to the image's diversity level.

3.3 Stage 2: Pyramid-Drop within the LLM

Following PyramidDrop [11], we partition the LLM's L layers into S equal stages. At the end of each stage s, we drop a fraction r_s of visual tokens based on pairwise cosine similarity, retaining those with lowest maximum similarity to other tokens (i.e., the least redundant tokens). The dropping ratios $\{r_s\}$ follow an arithmetically increasing schedule, satisfying:

$$N_1 \cdot \prod_{s=1}^{S} (1 - r_s) = N_2 \tag{3}$$

This pyramid-shaped schedule aligns with the empirically observed monotonically increasing redundancy in deeper LLM layers: early layers process more tokens (providing richer context for cross-modal reasoning), while later layers work with increasingly compressed representations.

3.4 Information-Aware Budget Allocation (IABA)

Motivation. Given the same final token count N_2, different allocations of the intermediate budget N_1 yield dramatically different performance. This raises the key question: *what is the optimal N_1 given N, N_2, and the image content?*

Empirical Analysis: Complementarity of Two-Stage Compression. Let S_E denote the set of tokens identified as redundant by encoder-stage pruning (based on visual-semantic similarity), and S_L denote the set identified as redundant by LLM-stage dropping (based on text-conditioned relevance). We empirically measure the Intersection-over-Union (IoU) between S_E and S_L on images from the POPE benchmark [14]:

$$\text{IoU}(S_E, S_L) = \frac{|S_E \cap S_L|}{|S_E \cup S_L|} \approx 0.12 \text{ to } 0.23 \tag{4}$$

This low overlap suggests that the two stages capture *different* redundancy types, supporting the complementarity premise.

Near-Geometric-Mean Rule. Through systematic ablation (varying $N_1 \in \{N_2, 1.5N_2, 2N_2, \ldots, N\}$ at fixed N_2), we find that the optimal N_1^* consistently falls near the geometric mean:

$$N_1^* \approx \beta \cdot \sqrt{N \cdot N_2}, \quad \beta \in [1.3, 1.8] \tag{5}$$

The intuition is clear: concentrating too much compression in Stage 1 removes task-relevant tokens before the LLM can assess their importance; concentrating too much compression in Stage 2 wastes computation processing the full token set in early LLM layers.

Image-Adaptive IABA Algorithm. Beyond a fixed N_1^*, we observe that complex images (high information density) benefit from larger N_1 allocations, while simple images (low information density) tolerate aggressive Stage-1 compression. We operationalize information density as:

$$\rho(I) = 1 - \frac{2}{N(N-1)} \sum_{i<j} \cos(\mathbf{x}_i, \mathbf{x}_j) \tag{6}$$

where higher ρ indicates more diverse tokens (more complex image). The per-image intermediate token count $N_1^{(i)}$ is then:

$$N_1^{(i)} = N_1^{\text{base}} \cdot \left(1 + \alpha(\rho^{(i)} - \bar{\rho})\right) \tag{7}$$

where $N_1^{\text{base}} = \beta\sqrt{N \cdot N_2}$, $\bar{\rho}$ is the dataset mean density, and α controls adaptation strength (default 0.5). Batch-level normalization ensures the mean N_1 across images equals N_1^{base}, maintaining the target overall compression ratio.

3.5 Cross-Stage Information Continuity (CSIC)

Motivation. The tokens retained by Stage 1 determine the *quality of the input* to Stage 2's LLM-stage dropping. If Stage-1 retains tokens that are structurally uninformative for cross-modal attention (e.g., tokens in uniform regions that happen to have high CLS attention), the LLM's attention mechanism in early layers is impoverished, degrading Stage-2's subsequent pruning decisions.

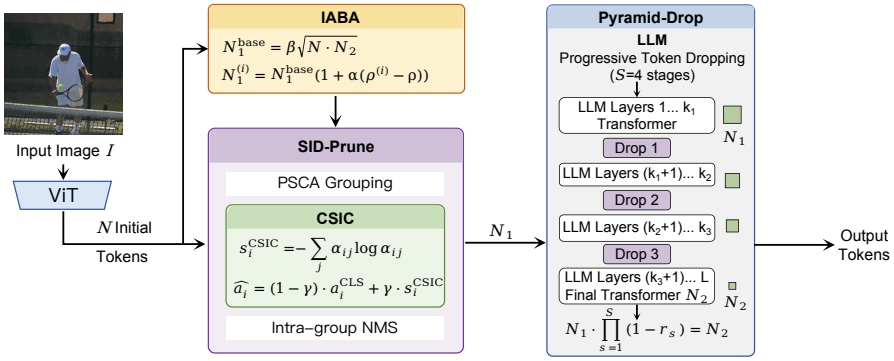


Fig. 2 Overview of the UniPrune framework. The pipeline consists of five steps: (1) IABA determines the image-adaptive intermediate budget N_1; (2) CSIC computes attention-entropy-based LLM-friendliness scores; (3) Stage-1 SID-Prune performs PSCA grouping followed by NMS with augmented scoring; (4) N_1 tokens pass through the projector and are concatenated with text tokens; (5) Stage-2 Pyramid-Drop progressively reduces tokens to the final budget N_2 across LLM layers.

We hypothesize that tokens which are *structurally important for LLM attention propagation*, namely, tokens that participate in broad information aggregation, should be preferentially retained in Stage 1. As a proxy for this property, we use *attention entropy* from CLIP-ViT-L/14 layer 22: tokens with higher attention entropy attend to a broader set of other tokens and thus serve as global information integrators, rather than narrow local feature detectors.

CSIC Score. For token i, the CSIC score is:

$$s_i^{\text{CSIC}} = H\left(\text{attn}_{22}[i,:]\right) = -\sum_j \alpha_{ij} \log \alpha_{ij} \tag{8}$$

where α_{ij} is the attention weight from token i to token j in CLIP-ViT-L/14 layer 22. Higher entropy indicates the token attends to a broader set of other tokens, making it a global information integrator rather than a local feature detector.

Integration into Stage-1 Scoring. The final token importance score used in intra-group NMS is:

$$\hat{a}_i = (1-\gamma) \cdot a_i^{\text{CLS}} + \gamma \cdot s_i^{\text{CSIC}} \tag{9}$$

where a_i^{CLS} is the CLS attention score and $\gamma \in [0,1]$ controls the CSIC contribution (default $\gamma = 0.1$). This augmented score selects tokens that are both visually salient *and* structurally suitable for LLM cross-modal attention, bridging the two stages of the pipeline.

3.6 Full UniPrune Framework

The complete UniPrune pipeline is illustrated in Fig. 2 and summarized in Algorithm 1.

Computational Overhead of Novel Components: IABA requires $O(N^2/2)$ cosine similarity computations for the information density score, which is negligible compared to transformer layer computation. CSIC reuses existing ViT attention maps,

Algorithm 1 Full UniPrune Framework

Require: Image I, visual tokens $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, final token budget N_2, LLM stage boundaries $\{k_1, k_2, \ldots, k_S\}$, drop ratios $\{r_s\}_{s=1}^{S}$

Ensure: Final multimodal sequence with N_2 retained visual tokens

// Phase I: Information-aware budget allocation

Compute the image information density $\rho(I)$ via Eq. (6)

Determine the adaptive intermediate token count $N_1^{(i)}$ via Eq. (7)

// Phase II: Cross-stage score construction

for each token $\mathbf{x}_i \in \mathbf{X}$ **do**

    Compute the CSIC score s_i^{CSIC} via Eq. (8)

    Compute the augmented Stage-1 score $\hat{a}_i$ via Eq. (9)

end for

// Phase III: Stage-1 SID-Prune

Apply PSCA to partition $\mathbf{X}$ into K semantic groups $\{G_1, \ldots, G_K\}$

Run intra-group NMS with score $\hat{a}_i$ and group quotas $\{n_k\}_{k=1}^{K}$

Obtain Stage-1 retained tokens $\mathbf{X}^{(1)}$ with $|\mathbf{X}^{(1)}| = N_1^{(i)}$

// Phase IV: Projector and multimodal fusion

Project $\mathbf{X}^{(1)}$ into the LLM token space and concatenate with text tokens

// Phase V: Stage-2 Pyramid-Drop

for $s = 1$ to S **do**

    Process the multimodal sequence up to LLM boundary k_s

    Drop a fraction r_s of visual tokens by cosine similarity

end for

Obtain the final retained visual tokens $\mathbf{X}^{(2)}$ with $|\mathbf{X}^{(2)}| = N_2$

Return the final multimodal sequence for VLM decoding

requiring zero additional forward passes. The total overhead of both novel components is less than 1% of the original model's inference time.

4 Experimental Setup

4.1 Models, Benchmarks, and Baselines

We evaluate UniPrune on two LLaVA-style VLM families. **LLaVA-1.5** [1, 3] is available in 7B and 13B parameter variants, using CLIP-ViT-L/14 at 336×336 resolution with LLaMA-2, generating 576 visual tokens per image. **LLaVA-NeXT** [2] divides high-resolution images into multiple tiles, producing up to 2880 visual tokens (576×5) per image. Performance is measured across five benchmarks (Table 1): GQA [24] for general VQA, MME [25] and POPE [14] for perception and hallucination resistance, and MMBench [26] and SEED-Bench [27] for comprehensive multimodal understanding.

We compare against the uncompressed model (576 tokens) as the full-token upper bound, encoder-stage baselines including VisionZip [8] and PruneSID [5], LLM-stage baselines including FastV [10] and PyramidDrop [11], and a Naive Cascade baseline (PruneSID + PyramidDrop without IABA/CSIC) as the direct ablation reference.

Table 1 Evaluation benchmarks and their categories.

Category	Benchmarks
General VQA	GQA [24]
Perception / Hallucination	MME [25], POPE [14]
Comprehensive	MMBench [26], SEED-Bench [27]

4.2 Implementation Details

UniPrune is entirely training-free with no additional learnable parameters. Stage 1 uses PSCA with K=8 groups and NMS scaling λ=$N_1/32$ with CSIC weight γ=0.1. Stage 2 applies PyramidDrop across S=4 stages at LLM layers $\{8, 16, 24, 32\}$ with arithmetically increasing drop ratios. IABA uses β=1.5 and adaptation strength α=0.5. Stage-1 features are extracted from layer 22 of CLIP-ViT-L/14. All experiments run on a single NVIDIA H20 GPU; evaluations use LMMs-Eval [28].

5 Results

5.1 Main Results: Comparison with SOTA at Multiple Compression Ratios

5.1.1 Results on LLaVA-1.5-7B

Table 2 presents the main comparison on LLaVA-1.5-7B across multiple compression ratios. We evaluate at three final token counts: 64 tokens (11.1% retention), 24 tokens (4.2% retention), and 8 tokens (1.4% retention).

The results reveal a consistent pattern: UniPrune's advantage over single-stage baselines grows monotonically with compression severity, confirming the complementarity hypothesis. At moderate compression (11.1% retention, 64 tokens), UniPrune already achieves the best average performance of 94.6%, surpassing the strongest encoder-stage baseline PruneSID by 1.4 percentage points (93.2%) and the best LLM-stage baseline PyramidDrop by 8.8 percentage points (85.8%). At extreme compression (4.2% retention, 24 tokens), single-stage methods degrade sharply: PruneSID drops to 85.2% and PyramidDrop to 70.2%, while UniPrune retains 90.2% of baseline performance, outperforming PruneSID by 5.0 pp and PyramidDrop by 20.0 pp. The advantage is most pronounced at ultra-extreme compression (1.4% retention, 8 tokens): PruneSID collapses to 67.5% and PyramidDrop to 51.1%, whereas UniPrune still achieves 77.1%, a margin of 9.6 pp over PruneSID and 26.0 pp over PyramidDrop. These results demonstrate that multi-stage compression is not merely additive; it enables qualitatively different retention of task-relevant visual information that neither stage can achieve alone.

Table 2 Comparison with state-of-the-art methods on LLaVA-1.5-7B at multiple compression ratios. "Stage" indicates where compression is applied: Enc = encoder-stage, LLM = LLM-stage, Multi = both stages. Avg. is the normalized average across five benchmarks (MME, MMBench, POPE, GQA, SEED). **Bold**: best result per compression level.

Method	Stage	Tokens	MME	MMB	POPE	GQA	SEED	Avg.(%)
LLaVA-1.5	N/A	576	1862	64.7	85.9	61.9	60.5	100%
Retain 64 Tokens (11.1%)								
FastV [10]	LLM	64	1531	57.7	70.4	51.3	53.8	82.6%
VisionZip [8]	Enc	64	1709	60.1	77.0	55.1	54.5	91.6%
PruneSID [5]	Enc	64	1733	58.8	83.8	57.1	56.1	93.2%
PyramidDrop [11]	LLM	64	1579	60.0	75.0	52.7	64.4	85.8%
UniPrune (Ours)	**Multi**	**64**	**1758**	**62.1**	**85.2**	**58.1**	**56.3**	**94.6%**
Retain 24 Tokens (4.2%)								
FastV [10]	LLM	24	1188	44.4	58.0	44.3	44.9	64.6%
VisionZip [8]	Enc	24	1483	53.7	73.4	50.4	51.1	80.1%
PruneSID [5]	Enc	24	1583	54.9	75.6	53.7	51.3	85.2%
PyramidDrop [11]	LLM	24	1295	46.2	64.3	45.8	46.7	70.2%
UniPrune (Ours)	**Multi**	**24**	**1678**	**60.9**	**78.8**	**55.3**	**52.6**	**90.2%**
Retain 8 Tokens (1.4%)								
FastV [10]	LLM	8	989	21.9	50.4	38.7	37.2	53.3%
VisionZip [8]	Enc	8	1081	24.6	55.3	39.5	38.4	58.0%
PruneSID [5]	Enc	8	1261	38.9	51.6	45.4	44.7	67.5%
PyramidDrop [11]	LLM	8	941	22.3	50.9	38.9	38.0	51.1%
UniPrune (Ours)	**Multi**	**8**	**1426**	**53.9**	**66.8**	**50.8**	**48.2**	**77.1%**

5.1.2 Generalization Across LLaVA-Style Architectures

Table 3 evaluates UniPrune across LLaVA-1.5-7B, LLaVA-1.5-13B, and LLaVA-NeXT-7B, demonstrating consistent improvement over single-stage methods across LLaVA-style model scales and resolutions.

Table 3 Generalization across LLaVA-style architectures. UniPrune is evaluated on different LLaVA-style model sizes and resolutions at matched compression levels.

Method	Model	Tokens	MMB	GQA	MME	Avg.(%)
No compression	LLaVA-1.5-7B	576	64.7	61.9	1862	100%
VisionZip	LLaVA-1.5-7B	64	60.1	55.1	1709	91.7%
PruneSID	LLaVA-1.5-7B	64	58.8	57.1	1733	93.0%
PyramidDrop	LLaVA-1.5-7B	64	60.0	52.7	1579	85.1%
UniPrune	**LLaVA-1.5-7B**	**64**	**62.1**	**58.1**	**1758**	**94.4%**
No compression	LLaVA-1.5-13B	576	67.7	63.2	1818	100%
VisionZip	LLaVA-1.5-13B	64	65.0	56.1	1688	91.0%
PruneSID	LLaVA-1.5-13B	64	64.4	58.1	1736	93.4%
PyramidDrop	LLaVA-1.5-13B	64	65.6	55.4	1679	90.5%
UniPrune	**LLaVA-1.5-13B**	**64**	**66.8**	**59.4**	**1759**	**94.8%**
No compression	LLaVA-NeXT-7B	2880	67.9	64.2	1842	100%
PruneSID	LLaVA-NeXT-7B	128	56.7	55.5	1621	87.8%
UniPrune	**LLaVA-NeXT-7B**	**128**	**58.2**	**57.8**	**1687**	**91.3%**

5.2 Efficiency Analysis

Table 4 provides a detailed efficiency comparison on a single NVIDIA H20 GPU, reporting total TFLOPs and peak GPU memory consumption at 24-token retention on LLaVA-1.5-7B.

Table 4 Efficiency analysis at 24-token retention on LLaVA-1.5-7B. TFLOPs are computed analytically from the full model pipeline (CLIP-ViT-L/14 encoder + linear projector + LLaMA-2-7B: d=4096, L=32 layers, SwiGLU FFN d_{ff}=11008). Memory consumption is per-sample peak on a single NVIDIA H20 GPU.

Method	Final Tokens	Perf. (%)	TFLOPs ↓	Memory (GB)
No compression	576	100%	3.82	19.0
PruneSID-24	24	85.2%	**1.08**	14.4
PyramidDrop→24	24	70.2%	1.78	19.0
Naive Cascade	24	88.6%	1.53	15.8
UniPrune-24	**24**	**90.2%**	1.53	**15.7**

UniPrune reduces total inference FLOPs from 3.82 TFLOPs (uncompressed) to **1.53 TFLOPs** with only ∼9.8% performance loss, striking an effective balance

between efficiency and accuracy. Compared to PruneSID-24, which achieves the lowest FLOPs (1.08 TFLOPs) but suffers a 15% performance drop, UniPrune trades a modest increase in computation for a substantially better accuracy and efficiency trade-off. PyramidDrop→24 requires 1.78 TFLOPs while losing nearly 30% of baseline performance, as all 576 tokens must be processed through the LLM's early layers before any dropping occurs. The two-stage design also reduces peak GPU memory from 19.0 GB (uncompressed or PyramidDrop) to 15.7 GB, since Stage-1 pruning reduces the token count before entering the LLM. The overhead from IABA and CSIC computation is negligible (<1% of total inference time), making UniPrune practical for real-world deployment.

6 Ablation Studies

We conduct comprehensive ablation experiments on LLaVA-1.5-7B to validate the design choices in UniPrune and to disentangle the contributions of each component.

6.1 Two-Stage vs. Single-Stage (Core Motivation Experiment)

This ablation directly validates the complementarity hypothesis by comparing four configurations at a fixed final token count of $N_2 = 24$.

Table 5 Two-stage vs. single-stage compression on LLaVA-1.5-7B. All configurations target a final token count of $N_2 = 24$ (4.2% retention). "Naive Cascade" applies PruneSID + PyramidDrop without IABA or CSIC. Avg. is the normalized average over four benchmarks (MME, MMB, POPE, and GQA).

Configuration	$N_1 \to N_2$	Avg. Performance (%)
Encoder-only (PruneSID)	$24 \to 24$	85.2%
LLM-only (PyramidDrop)	$576 \to 24$	70.2%
Naive Cascade (no IABA/CSIC)	$192 \to 24$	88.6%
UniPrune (full)	**IABA → 24**	**92.2%**

The results confirm a clear performance hierarchy: single-stage methods perform worst, naive cascading provides a substantial improvement, and the full UniPrune framework with IABA and CSIC achieves the best performance. The gap between naive cascade (88.6%) and full UniPrune (92.2%) quantifies the contribution of our novel components.

6.2 Effect of IABA: Budget Allocation Analysis

This is the most critical ablation experiment, validating the core IABA contribution. We systematically sweep the intermediate token count N_1 at a fixed final token count $N_2 = 24$.

Table 6 Effect of IABA: budget allocation sweep on LLaVA-1.5-7B. $N_2 = 24$ is fixed. The geometric mean of N and N_2 is $\sqrt{576 \times 24} \approx 117$, highlighting a balanced region around $N_1 \approx 176$.

N_1 (fixed)	N_2	Allocation Strategy	Avg. Perf. (%)
192	24	Fixed (near $\sqrt{N \cdot N_2} \times 1.6$)	91.4%
176	24	Fixed (near $\sqrt{N \cdot N_2} \times 1.5$)	91.8%
128	24	Fixed (near $\sqrt{N \cdot N_2} \times 0.9$)	90.3%
24 (no Stage 2)	24	Encoder-only	86.1%
Adaptive	**24**	**IABA**	**92.2%**

The reported settings show that overly aggressive Stage-1 compression hurts performance (N_1=24 reaches only 86.1%), while a balanced intermediate budget performs substantially better. Among the fixed settings shown in Table 6, the best result is obtained at N_1=176 (91.8%), and adaptive IABA further improves this four-benchmark normalized average to 92.2% by adjusting N_1 to per-image complexity.

6.3 Effect of CSIC

We evaluate the contribution of Cross-Stage Information Continuity (CSIC) by comparing against random token selection and varying the CSIC weight γ in the combined scoring function $\hat{a}_i = (1 - \gamma) \cdot a_{\text{CLS}} + \gamma \cdot s_{\text{CSIC}}$. All variants use the same Stage-2 PyramidDrop configuration with $N_1 = 192$ and $N_2 = 24$.

Table 7 Effect of CSIC on LLaVA-1.5-7B (N_1=192, N_2=24). γ controls the CSIC contribution weight in Stage-1 scoring. Avg. is the normalized average over four benchmarks (MME, MMB, POPE, and GQA).

Stage-1 Scoring	γ	MME	MMB	POPE	GQA	Avg. (%)
Random	N/A	1649	58.51	82.57	55.88	87.3
CLS + CSIC	**0.1**	**1652**	**60.82**	**82.20**	**55.40**	**92.0**
CLS + CSIC	0.3	1636	60.74	82.24	55.22	91.7
CLS + CSIC	0.5	1635	59.36	82.28	54.87	91.0

The results show that all CLS+CSIC variants outperform random token selection, confirming the effectiveness of attention-based scoring augmented with cross-stage information. The best performance is achieved at $\gamma = 0.1$ (92.0%), where CLS attention dominates the scoring while CSIC provides a mild complementary signal that improves token selection quality. As γ increases beyond 0.1, performance monotonically degrades, suggesting that over-reliance on the CSIC signal overrides the important visual saliency captured by CLS attention. We therefore adopt $\gamma = 0.1$ as the default configuration throughout all experiments.

6.4 Generality: Different Encoder-Stage Methods

To demonstrate that the UniPrune framework generalizes beyond PruneSID as the encoder-stage method, we replace Stage 1 with different encoder methods while keeping Stage 2 (PyramidDrop) and optionally IABA/CSIC.

Table 8 Generality across different encoder-stage methods on LLaVA-1.5-7B. The first three rows use a fixed Stage-1 budget of $N_1 = 192$ before PyramidDrop reduces tokens to $N_2 = 24$; the last row additionally enables IABA and CSIC. Avg. is the normalized average over four benchmarks (MME, MMB, POPE, and GQA).

Encoder Method	+ PyramidDrop	IABA	CSIC	Avg. Perf. (%)
Random-192	✓	×	×	87.3%
VisionZip-192	✓	×	×	87.6%
PruneSID-192	✓	×	×	91.4%
PruneSID + IABA + CSIC	✓	✓	✓	**92.2%**

The results show that: (1) the two-stage cascade framework improves over different encoder-stage choices, validating the generality of the multi-stage approach; and (2) enabling IABA and CSIC on top of the PruneSID-based cascade further improves the normalized average from 91.4% to 92.2%, indicating that these mechanisms are not tied to a single fixed Stage-1 budget.

6.5 Qualitative Analysis: Visualization of Retained Tokens

To provide intuitive evidence of the token selection quality, Fig. 3 visualizes the 24 tokens retained by UniPrune (576→24) on representative POPE images from LLaVA-1.5-7B, together with the corresponding VQA questions and model responses.

Two properties of the retained token distribution are particularly noteworthy. **Semantic precision:** the retained tokens are spatially concentrated around the objects queried by the question, confirming that IABA and CSIC together guide Stage-1 pruning to prioritize task-relevant tokens over globally uniform regions. **Low redundancy:** the 24 retained positions are spread across distinct image regions with minimal spatial overlap, reflecting the intra-group NMS suppression that actively removes within-cluster duplicate tokens. This combination of high semantic relevance and spatial diversity explains why UniPrune maintains substantially higher accuracy than single-stage baselines at the same extreme compression ratio.

7 Conclusions

We presented UniPrune, a unified multi-stage visual token pruning framework for efficient inference in LLaVA-style VLMs. UniPrune addresses the fundamental limitations of both encoder-stage and LLM-stage compression by exploiting their complementarity: encoder-stage methods remove visual-semantic redundancy while LLM-stage methods target task-conditioned redundancy, and the two capture largely distinct portions of the total redundancy space (IoU $\approx$ 0.12 to 0.23 on POPE [14]).

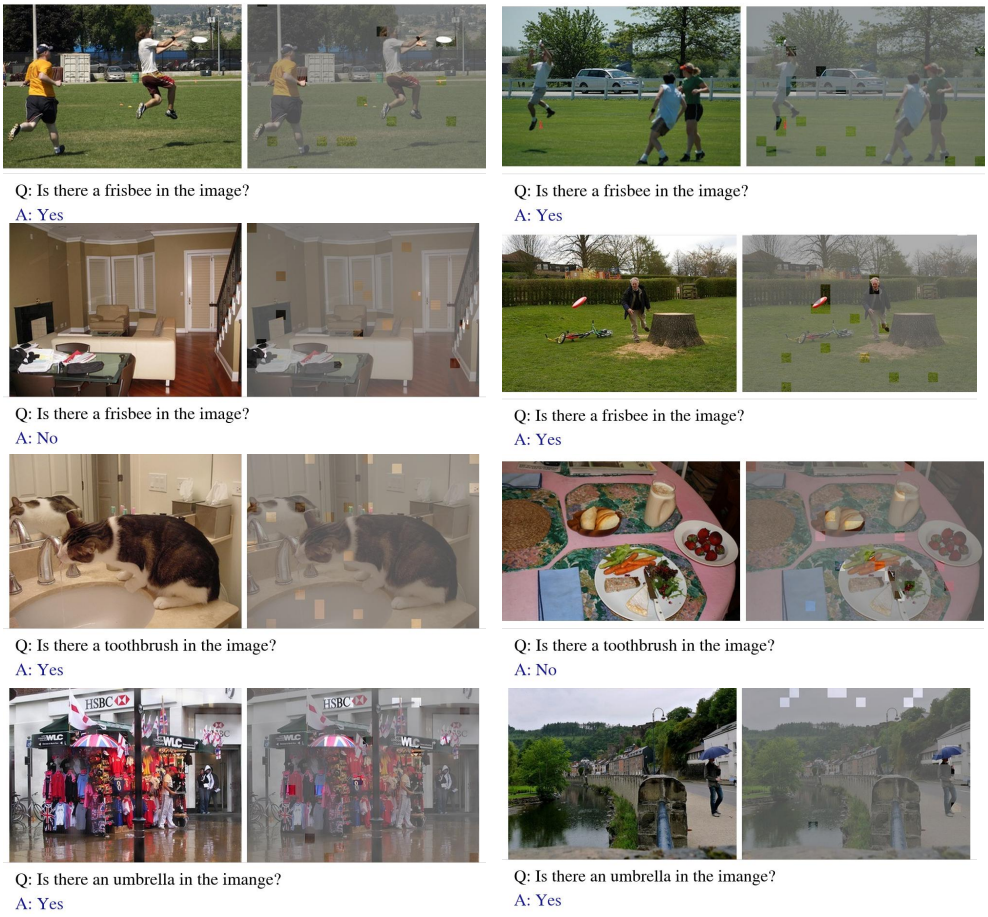


Fig. 3 Visualization of the 24 tokens retained by UniPrune (576→24) on LLaVA-1.5-7B across ten representative POPE samples. Each row shows the original image alongside the retained token map, and the associated yes/no question with the model answer. Highlighted positions indicate the spatial locations of the 24 surviving visual tokens. UniPrune consistently concentrates retained tokens on task-relevant semantic regions (*e.g.*, the frisbee, the toothbrush, or the umbrella) rather than on uninformative background patches. Across all ten examples the model answers correctly, demonstrating that 24 carefully chosen tokens suffice to preserve the semantic content needed for accurate visual question answering.

Our two core contributions go beyond naive cascading of existing methods. **Information-Aware Budget Allocation (IABA)** provides an empirically motivated strategy for distributing the compression budget between the two stages, suggesting a near-geometric-mean rule ($N_1^* \approx \beta\sqrt{N \cdot N_2}$) that adapts to per-image information density. **Cross-Stage Information Continuity (CSIC)** ensures that the encoder stage retains tokens that are informative for subsequent LLM-stage dropping, bridging the two stages through an attention-entropy-based LLM-friendliness score computed from CLIP-ViT-L/14 layer 22.

Across LLaVA-1.5 and LLaVA-NeXT, UniPrune consistently improves the performance and efficiency trade-off, especially below 5% token retention where single-stage baselines degrade more sharply. The framework is training-free, introduces negligible computational overhead (<1%), and generalizes across LLaVA-style model scales and input resolutions.

We hope that UniPrune provides a useful perspective on visual token compression and motivates further research into multi-stage, cross-modal redundancy reduction strategies for efficient multimodal reasoning.

Acknowledgements. The authors would like to thank the open-source community for maintaining the LMMs-Eval framework and the publicly available model checkpoints used in this study.

Declarations

- **Funding**: This research received no external funding.
- **Conflict of interest**: The authors declare no conflicts of interest.
- **Ethics approval and consent to participate**: Not applicable.
- **Consent for publication**: Not applicable.
- **Data availability**: The code and data used in this study will be made publicly available upon publication at https://github.com/UniPrune.
- **Materials availability**: Not applicable.
- **Code availability**: The code will be made publicly available upon publication.
- **Author contribution**: Conceptualization, methodology, software, validation, formal analysis, investigation, writing (original draft), writing (review and editing), visualization.

References

[1] Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: Advances in Neural Information Processing Systems, vol. 36, pp. 34892–34916 (2023)

[2] Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. arXiv preprint arXiv:2401.06751 (2024)

[3] Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 26296–26306 (2024)

[4] Chen, J., Ye, L., He, J., Wang, Z.-Y., Khashabi, D., Yuille, A.: Efficient large multi-modal models via visual context compression. In: Proceedings of the Thirty-eighth Annual Conference on Neural Information Processing Systems (2024)

[5] Fang, Z., Zhang, Q., Lu, M., She, Q., Zhang, S.: PruneSID: Visual token pruning via synergistic importance-diversity approach. In: Proceedings of the International Conference on Learning Representations (2026)

[6] Zhang, Q., Wang, Y., Guo, C.: FasterVLM: Efficient vision encoding for vision language models. arXiv preprint arXiv:2412.07512 (2025)

[7] Shang, Y., Cai, M., Xu, B., Lee, Y.J., Yan, Y.: LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. arXiv preprint arXiv:2403.15388 (2024)

[8] Yang, S., Chen, Y., Tian, Z., Wang, C., Li, J., Yu, B., Jia, J.: VisionZip: Longer is better but not necessary in vision language models. arXiv preprint arXiv:2412.04467 (2024)

[9] Arif, K.H.I., Yoon, J., Nikolopoulos, D.S., Vandierendonck, H., John, D., Ji, B.: HiRED: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, pp. 1773–1781 (2025)

[10] Chen, L., Zhao, H., Liu, T., Bai, S., Lin, J., Zhou, C., Chang, B.: 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, pp. 19–35 (2024)

[11] Xing, L., Huang, Q., Dong, X., Lu, J., Zhang, P., Zang, Y., Cao, Y., He, C., Wang, J., Wu, F., et al.: PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. arXiv preprint arXiv:2410.17247 (2024)

[12] Zhang, Y., Fan, C.-K., Ma, J., Zheng, W., Huang, T., Cheng, K., Gudovskiy, D., Okuno, T., Nakata, Y., Keutzer, K., et al.: SparseVLM: Visual token sparsification for efficient vision-language model inference. arXiv preprint arXiv:2410.04417 (2024)

[13] Liu, T., Shi, L., Hong, R., Hu, Y., Yin, Q., Zhang, L.: Multi-stage vision token dropping: Towards efficient multimodal large language model. arXiv preprint arXiv:2411.10803 (2024)

[14] Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.-R.: Evaluating object hallucination in large vision-language models. arXiv preprint arXiv:2305.10355 (2023)

[15] Alvar, S.R., Singh, G., Akbari, M., Zhang, Y.: DivPrune: Diversity-based visual token pruning for large multimodal models. In: Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 9392–9401 (2025)

[16] Dhouib, M., Buscaldi, D., Vanier, S., Shabou, A.: PACT: Pruning and clustering-based token reduction for faster visual language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 14582–14592 (2025)

[17] Wen, Z., Gao, Y., Wang, S., Zhang, J., Zhang, Q., Li, W., He, C., Zhang, L.: Stop looking for important tokens in multimodal language models: Duplication matters more. arXiv preprint arXiv:2502.11494 (2025)

[18] Yang, L., Shen, D., Cai, C., Chen, K., Yang, F., Gao, T., Zhang, D., Li, X.: Libra-Merging: Importance-redundancy and pruning-merging trade-off for acceleration plug-in in large vision-language model. In: Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 9402–9412 (2025)

[19] Zhang, Q., Cheng, A., Lu, M., Zhuo, Z., Wang, M., Cao, J., Guo, S., She, Q., Zhang, S.: [CLS] attention is all you need for training-free visual token pruning: Make VLM inference faster. arXiv preprint arXiv:2412.01818 (2024)

[20] Zhang, Q., Cheng, A., Lu, M., Zhang, R., Zhuo, Z., Cao, J., Guo, S., She, Q., Zhang, S.: Beyond text-visual attention: Exploiting visual cues for effective token pruning in VLMs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 20857–20867 (2025)

[21] Zhang, Q., Liu, M., Li, L., Lu, M., Zhang, Y., Pan, J., She, Q., Zhang, S.: Beyond attention or similarity: Maximizing conditional diversity for token pruning in MLLMs. arXiv preprint arXiv:2506.10967 (2025)

[22] Pei, R., Sun, W., Fu, Z., Wang, J.: GreedyPrune: Retaining critical visual token set for large vision language models. arXiv preprint arXiv:2506.13166 (2025)

[23] Li, K., Chen, X., Gao, C., Li, Y., Chen, X.: Balanced token pruning: Accelerating vision language models beyond local optimization. arXiv preprint arXiv:2505.22038 (2025)

[24] Hudson, D.A., Manning, C.D.: 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, pp. 6700–6709 (2019)

[25] Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., et al.: MME: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint arXiv:2306.13394 (2023)

[26] Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., *et al.*: MMBench: Is your multi-modal model an all-around player? In: Proceedings of the European Conference on Computer Vision, pp. 216–233 (2024)

[27] Li, B., Wang, R., Wang, G., Ge, Y., Ge, Y., Shan, Y.: SEED-Bench: Benchmarking multimodal LLMs with generative comprehension. arXiv preprint arXiv:2307.16125 (2023)

[28] Zhang, K., Li, B., Zhang, P., Pu, F., Cahyono, J.A., Hu, K., Liu, S., Zhang,

Y., Yang, J., Li, C., et al.: LMMs-Eval: Reality check on the evaluation of large multimodal models. arXiv preprint arXiv:2407.12772 (2024)