HAP: Head-Adaptive Visual Token Pruning via Cross-Modal Alignment

Anonymous ACL submission

Abstract

Recent Vision-Language Models encode high-resolution images into long visual token sequences, incurring prohibitive prefill costs. To compress them, existing methods score each visual token by averaging text-to-visual attention uniformly across all heads, which assumes every head matches the query. However, our empirical analysis shows that misaligned heads dominate the average, amplifying background tokens and drowning out fine-grained cues.

To address this, we propose PAQ (Prompt-Grounded Attention Quality), a metric quantifying how well each head aligns the prompt with image regions. Built on PAQ, our pruning proceeds in three stages. Given a target FLOPs budget, we first partition the transformer layers into groups and allocate a visual token budget to each. Within each group, we then aggregate per-head attention maps via PAQ-weighted softmax into a group-level matrix. Finally, we score visual tokens by this matrix's magnitude and retain the allocated budget per group. By weighting heads with PAQ, our method scores tokens by true relevance to the prompt rather than diluting it through uniform averaging.

Across 18 benchmarks, our method delivers state-of-the-art trade-offs. Specifically, on LLaVA-1.5-7B (9 tasks), retaining only **5.6%** tokens preserves **99.1%** of the original performance, surpassing the strongest baseline AutoPrune by 4.2 points. Code is available.

1 Introduction

Recent advances in Vision-Language Models (VLMs) (Liu et al., 2023; Lu et al., 2025; Bai et al., 2025) have achieved remarkable progress in multimodal reasoning and understanding. To capture fine-grained details, state-of-the-art VLMs increasingly adopt high-resolution visual encoders that produce long sequences of visual tokens. This visual token explosion imposes substantial computational and memory overhead during the autoregressive prefill stage, creating a critical bottleneck for real-time deployment and limiting the scalability of VLMs in resource-constrained environments.

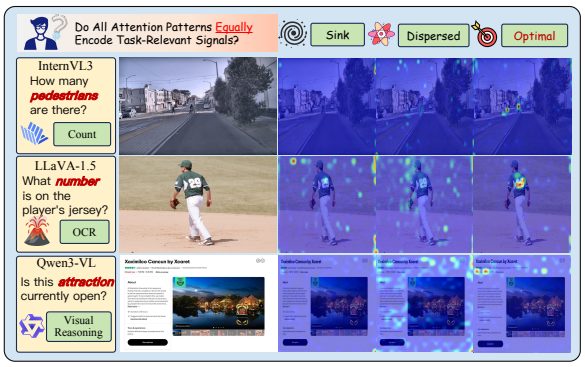

Figure 1: **Head-level cross-modal attention patterns.** Layer 2 attention maps reveal *sink*, *optimal*, and *dispersed* cross-modal patterns across three VLMs.

To mitigate this prefill burden, existing methods exploit cross-modal attention statistics to rank and prune visual tokens. These weights are already computed in the forward pass and naturally indicate where the model focuses under the text prompt. For instance, SparseVLM (Zhang et al., 2024b) identifies informative text tokens as raters and accumulates their attention mass over visual tokens to produce a layer-wise importance score. PDrop (Xing et al., 2024) aggregates attention scores uniformly across all heads within each target layer and progressively discards low-ranking tokens following a pyramid schedule. AutoPrune (Wang et al.) further derives mutual information from attention distributions to guide layer-wise budget allocation.

Despite their varied designs, these methods share a critical simplification. They implicitly assume that all attention heads within a layer exhibit the same degree of focus on prompt-relevant regions. Consequently, they average attention scores indiscriminately across all heads. However, our empirical analysis in Fig. 1 reveals attention heads split into three distinct modes (sink, dispersed, and optimal) with starkly different spatial focus. Sink

heads collapse to fixed regions regardless of the text prompt. Dispersed heads spread attention broadly without clear prompt correspondence. Only optimal heads concentrate on task-relevant regions that shift with the query. This reveals a critical yet overlooked property. The image regions that the model attends to under text guidance vary drastically across heads. We term this property *prompt-grounded attention*. Fig. 2 makes this cost concrete, showing that tokens selected by uniform head averaging consistently fall short of those selected by the single best-aligned head, both in task accuracy and in spatial localization.

To distinguish reliable prompt-grounded attention from spurious noise, we propose the **Prompt-Grounded Attention Quality (PAQ)** score. Based on the uncertainty coefficient (Theil, 1970), PAQ is computed purely from post-softmax intrinsic attention statistics. It measures how genuinely an attention head's visual distribution is driven by the text prompt. A high PAQ indicates concentrated attention on semantically prompt-relevant regions, while a low PAQ reflects attention-sink or dispersed patterns that are unreliable for pruning.

Building upon PAQ, we introduce **HAP**, a head-adaptive visual token pruning framework without any hyper-parameter tuning. Given a target FLOPs budget, HAP organizes transformer layers into groups and allocates visual token budgets following a geometric pyramid structure, progressively reducing tokens from lower to higher layers. Within each layer, HAP evaluates every attention head with PAQ. Heads with high PAQ, which concentrate on prompt-relevant regions, receive greater weight in the scoring. Heads with low PAQ are effectively softly suppressed. This produces a quality-aware layer-level score. HAP then hierarchically merges these layer scores across the group under the same PAQ weighting to obtain the final token importance ranking. This two-stage design replaces uniform head averaging with a principled mechanism that scores visual tokens by their true prompt relevance.

We validate the effectiveness of our method across 18 benchmarks and four model architectures, including LLaVA, InternVL, Qwen-VL and Deepseek-VL. In terms of accuracy, retaining only 5.6% of the visual tokens, our model still preserves 99.1% of the original performance, outperforming the strongest baseline AutoPrune by 4.2 points. In terms of efficiency, it reduces KV-cache usage by 80% and achieves a 2.82× inference speedup. Our contributions are summarized as follows:

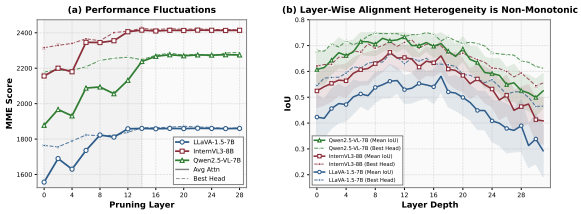

Figure 2: (a) MME performance when retaining 25% of visual tokens at different layers across three VLMs. (b) RefCOCO IoU when retaining 50% of visual tokens at different layers. In both panels, solid curves denote average attention mass and dashed curves denote the best single head; average aggregation consistently trails the optimal head.

- We empirically reveal diverse cross-modal alignment patterns across attention heads. We propose PAQ, a metric that discriminates reliable alignment from attention-sink or dispersed patterns in a single score.
- We propose HAP, a training-free pruning strategy without hyper-parameter. HAP replaces coarse head averaging with PAQ-guided softmax aggregation, ensuring visual tokens are scored by their true relevance to the prompt.
- We evaluate HAP on five representative VLM architectures across 18 benchmarks. Results show that HAP achieves state-of-the-art performance-efficiency trade-offs, validating the strong generalizability of our approach.

2 Related Work

Vision-Language Models. Recent large vision-language models (VLMs) have achieved remarkable multimodal reasoning by integrating high-resolution visual encoders with LLMs (Hurst et al., 2024; Liu et al., 2023; Guo et al., 2025; Bai et al., 2025; Sun et al., 2026a). However, these encoders generate dense visual token sequences—ranging from hundreds to thousands per image—creating substantial computational and memory bottlenecks during prefill (Liu et al., 2024a; Chen et al., 2024c). This *visual token explosion* critically hinders efficient deployment of modern VLMs.

Visual Token Pruning for VLMs. Visual token pruning approaches fall into training-based and training-free paradigms. Training-based methods redesign the visual encoder or inject learnable compression modules (Xing et al., 2024; Cha et al., 2024; Zhang et al., 2025c), requiring expensive retraining and lacking scalability across diverse

backbones. Training-free methods avoid retraining by leveraging unimodal visual attention or cross-modal attention matrices. Visual-modal approaches (Arif et al., 2025; Zhang et al., 2025b) select tokens based on visual saliency but inherently lack textual guidance, often discarding task-relevant background. Cross-modal methods (Zhang et al., 2024b; Wang et al.; Zhang et al., 2025a) exploit vision-language interactions but typically aggregate attention heads uniformly and apply pruning at pre-defined layers, failing to exploit the heterogeneous cross-modal alignment quality across different model components.

3 Method

In this section, we formalize the VLM pruning objective and identify two critical limitations of strategies: uniform head aggregation and fixed-layer pruning (Sec. 3.1). We then propose the **Prompt-Grounded Attention Quality** (PAQ) score to discriminate reliable cross-modal alignment from misaligned and dispersed patterns (Sec. 3.2). Building upon PAQ, we introduce **HAP** (Sec. 3.3). Finally, we conclude with theoretical analysis of computational efficiency and system compatibility (Sec. 3.4). An overview is shown in Fig. 3.

3.1 Rethinking Token Pruning

VLMs generate textual responses conditioned on visual inputs and text prompts. An image is encoded into visual tokens $\mathbf{V} \in \mathbb{R}^{N_v \times D}$, accompanied by a system prompt $\mathbf{S} \in \mathbb{R}^{N_s \times D}$ and a user query $\mathbf{T} \in \mathbb{R}^{N_t \times D}$. The model auto-regressively generates a response $\mathbf{R}$ by modeling $p(\mathbf{R} \mid \mathbf{S}, \mathbf{V}, \mathbf{T})$.

Token pruning is formalized as a query-conditioned relevance ranking problem at the prefill stage, retaining only the visual tokens most relevant to the text query. Previous methods (Zhang et al., 2024b; Wang et al., 2025) compute a cross-modal score $s(\mathbf{v}_i)$ for each token and retain the top-K:

$$\mathbf{V}' = \{\mathbf{v}_i \mid s(\mathbf{v}_i) \in \text{top-}K(\{s(\mathbf{v}_j)\}_{j=1}^{N_v})\}, \quad (1)$$

where K is a pre-defined budget. Concretely, these methods aggregate attention scores uniformly across all H heads in a fixed layer k:

$$s(\mathbf{v}_i) = \frac{1}{H} \sum_{h=1}^{H} \mathbf{A}_h^k(\mathbf{v}_i, \mathbf{T}), \quad (2)$$

where $\mathbf{A}_h^k(\mathbf{v}_i, \mathbf{T})$ measures the total attention mass from text to visual token $\mathbf{v}_i$ in head h.

However, Eq. (2) assumes all attention heads contribute equally to prompt-grounded visual focus, which fails in practice. As Fig. 1 illustrates, heads within a single layer exhibit heterogeneous patterns (sink, dispersed, and optimal), rendering uniform aggregation suboptimal. These patterns also manifest across layers and architectures.

We conduct two complementary experiments to quantify the cost of ignoring this head-level variance. In Fig. 2a, we retain 25% of visual tokens at each layer independently and measure MME (Zhang et al., 2024a) performance across three VLMs. In Fig. 2b, we retain 50% of tokens at each layer on RefCOCO (Kazemzadeh et al., 2014) and measure IoU. Both figures consistently reveal the same gap: **visual tokens selected by the average attention mass consistently trail those selected by the best single head** in both task performance and spatial precision. Uniform aggregation therefore dilutes the fine-grained signals from optimal heads while amplifying noise from sink and dispersed ones.

Together, the layer-wise fluctuation and head-wise gap demonstrate that token pruning requires selective exploitation of prompt-grounded attention rather than uniform aggregation.

3.2 Prompt-Grounded Attention Quality

To select heads that encode reliable visual focus driven by the text prompt, we define the **Prompt-Grounded Attention Quality (PAQ)** score. Following the *uncertainty coefficient* (Theil, 1970), PAQ is formulated as:

$$\text{PAQ} = \frac{I(\mathbf{V}; \mathbf{T})}{H(\mathbf{V})}, \quad (3)$$

where $I(\mathbf{V}; \mathbf{T})$ measures how much the visual distribution is driven by the text query, and $H(\mathbf{V})$ measures total visual uncertainty. This ratio, bounded between 0 and 1, quantifies the proportion of visual distribution explained by the query.

We compute PAQ purely from intrinsic attention statistics, without extra parameters. Following prior work (Vaswani et al., 2017; Wang et al.), we consider the cross-modal attention map that encodes text-to-vision alignment. Let α_{ji} denote the normalized attention weight from the j-th text token t_j to the i-th visual token v_i in a given head. We interpret this cross-modal weight as a soft relevance assignment $\hat{p}(v_i \mid t_j) = \alpha_{ji}$, while adopting a uniform text prior $p(t_j) = 1/N_t$. This yields the joint distribution $p(v_i, t_j) = \frac{1}{N_t}\alpha_{ji}$ and

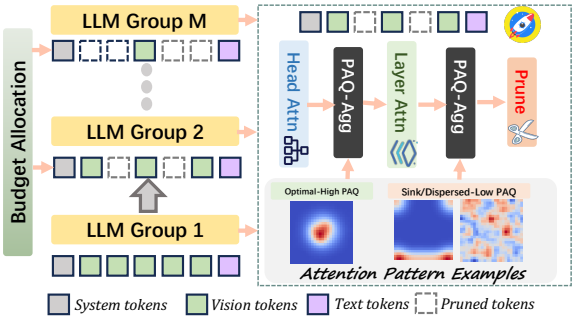

Figure 3: **Overview of HAP.** Left: FLOPs-driven budget allocation progressively reduces visual tokens across M layer groups. Right: unified PAQ-softmax aggregation. Head attention maps are softly fused within each layer and then aggregated within each group, yielding a reliable score for visual token pruning.

marginal $p(v_i) = \sum_j p(v_i, t_j)$. Substituting these into Eq. (3) gives the PAQ score:

$$\begin{aligned} I(\mathbf{V};\mathbf{T}) &= \sum_{i,j} p(v_i,t_j) \log \frac{p(v_i,t_j)}{p(v_i)p(t_j)}, \\ H(\mathbf{V}) &= -\sum_i p(v_i) \log p(v_i). \end{aligned} \tag{4}$$

Consequently, the three failure modes from Fig. 1 map directly onto this formulation. For both **sink** and **dispersed** heads, the attention weights α_{ji} are invariant to the text token t_j. In a sink head, every text token attends to the same small set of visual tokens. In a dispersed head, every text token spreads attention uniformly over all visual tokens. In either case, the conditional probability $\hat{p}(v_i \mid t_j) = \hat{p}(v_i)$ for all j, meaning the visual distribution is independent of the text token. Then, the joint distribution factorizes as $p(v_i, t_j) = p(v_i)p(t_j)$ and $I(\mathbf{V};\mathbf{T}) = 0$. Consequently, the PAQ numerator vanishes for both modes, suppressing them to near-zero.

Only **optimal** heads exhibit query-conditioned sharp peaks where $\hat{p}(v_i \mid t_j)$ varies with t_j, concentrating probability mass on task-relevant visual tokens. This yields $I(\mathbf{V};\mathbf{T}) > 0$ and keeps $H(\mathbf{V})$ low by concentrating attention on a small token subset. The resulting high PAQ score uniquely distinguishes optimal heads.

3.3 HAP

Building upon the PAQ score (Eq. (3)), we present HAP, a **h**ead-**a**daptive visual token **p**runing framework without any hyper-parameter tuning.

HAP first partitions the L transformer layers into M consecutive groups. Each group has L/M layers. Prior works (Xing et al., 2024) observe that visual tokens naturally follow a pyramid structure, with token volume declining from lower to higher layers as features consolidate. HAP inherits this observation and formalizes it through a geometric decay: the target budget for group g is $N_v(g) = V/(1+g)^2$, where V is the initial number of visual tokens. Consequently, the stage count M is derived directly from the target FLOPs budget via a closed-form cost function $\mathrm{F}(M)$ (Eq. 10), eliminating the need for per-task tuning.

Then, HAP progressively aggregates optimal heads with prompt-grounded attention in groups through a PAQ-guided weighting scheme. Consider a candidate set $\mathcal{C}$, where each element $c \in \mathcal{C}$ represents cross-modal attention, and is associated with PAQ score PAQ_c. To derive aggregation weights, we center the raw scores by subtracting the mean:

$$s_c = \mathrm{PAQ}_c - \frac{1}{|\mathcal{C}|} \sum_{j \in \mathcal{C}} \mathrm{PAQ}_j, \tag{5}$$

where s_c denotes the centered score for candidate c. We then apply standard softmax to obtain the aggregation weight:

$$w_c = \frac{\exp(s_c)}{\sum_{j \in \mathcal{C}} \exp(s_j)}. \tag{6}$$

This yields a soft, self-normalized distribution. Candidates with higher PAQ naturally dominate, while suboptimal ones are down-weighted. The aggregated attention map is then computed as:

$$\mathbf{A}_{\text{agg}} = \sum_{c \in \mathcal{C}} w_c \cdot \mathbf{A}_c. \tag{7}$$

Based on this, HAP first aggregates all heads within layer l. For layer l, we compute a PAQ score for each head from its attention map $\mathbf{A}_h^{(l)}$. Treating these H heads as a candidate set, we obtain weights via Eq. (5)–(6) and fuse their maps to produce the layer attention $\mathbf{A}_{\text{agg}}^{(l)}$. HAP then aggregates these layer attentions within each group. Next, we treat L/M layers of each group as a new candidate set. Each candidate has an aggregated attention map $\mathbf{A}_{\text{agg}}^{(l)}$ and a PAQ score. We apply the same PAQ-softmax procedure. Fusing these layer maps yields the final group attention $\mathbf{A}_{\text{group}}^{(g)}$ for token scoring, effectively amplifying signals from optimal heads.

Unlike prior methods that select fixed anchor layers and average all heads uniformly, HAP softly aggregates optimal heads in groups. These cross-modal attentions ensure that visual tokens are scored by their true relevance to the prompt.

Finally, we compute an importance score for each visual token by averaging over all text positions for group g:

$$s_i^{(g)} = \frac{1}{N_t} \sum_{j=1}^{N_t} \mathbf{A}_{\text{group}}^{(g)}(j, i). \tag{8}$$

We sort visual tokens by $s_i^{(g)}$ and retain exactly $N_v(g)$ tokens with the highest scores throughout the prefill phase.

3.4 Efficiency and Compatibility

Following PDrop (Xing et al., 2024), the FLOPs of a transformer layer is:

$$\mathrm{F}(N_v) \approx 4N_vD^2 + 2N_v^2D + 3N_vDd_{\text{ffn}}, \tag{9}$$

where N_v, D, and d_{ffn} denote the visual token count, hidden dimension, and FFN intermediate dimension, respectively.

For PAQ computation and token pruning, we directly reuse the attention matrices from the forward pass. Concretely, computing the PAQ score for one head requires marginal aggregation over $N_t \times N_v(l)$ attention weights, incurring $\mathcal{O}(N_tN_v(l))$ operations; with H heads per layer, the per-layer PAQ cost is $\mathcal{O}(HN_tN_v(l))$. Head aggregation via Eq. (5)–(6) adds $\mathcal{O}(H)$ and is absorbed into this term. Token selection averages the aggregated attention over N_t text positions and sorts $N_v(l)$ visual tokens, contributing $\mathcal{O}(N_tN_v(l) + N_v(l)\log N_v(l))$. Layer aggregation within each group repeats the PAQ-softmax over L/m layers, adding $\mathcal{O}((L/m)N_tN_v(l))$ per group. Because $N_v(l)$ decays geometrically with depth, summing these components across all L layers bounds the total overhead by $\mathcal{O}(HN_tVL/m)$. This remains strictly smaller than the unpruned attention cost and, under typical configurations where $D \gg H$ and $V \gg N_t$, constitutes a lower-order term relative to the full forward-pass FLOPs. Specifically, we extend the dual-flash strategy from SparseVLM (Zhang et al., 2024b) to compute per-head cross-modal distributions without materializing the full $\mathcal{O}(N^2)$ attention map (see Appendix B). Taken together with the pruned transformer costs, the total computational cost is fully characterized by the following closed-form function of the stage count m (see Appendix A for the full derivation):

$$\begin{aligned}\mathrm{F}_{\text{pruned}}(m) \approx \frac{L}{m}\Bigg[A\left(1 - \frac{1}{1+m}\right) \\ + \frac{B}{3}\left(1 - \frac{1}{(1+m)^3}\right)\Bigg],\end{aligned} \tag{10}$$

where $A = 4VD^2 + 3VDd_{\text{ffn}}$ and $B = 2V^2D$ are architecture constants. Equation (10) confirms that the cost is a deterministic function of m alone; no architecture-specific pruning schedule or per-task calibration is required.

4 Experiments

4.1 Experimental Settings

To assess the effectiveness of our method, we conduct experiments on LLaVA-1.5-7B, Qwen2.5-VL-7B, Qwen3-VL-8B, InternVL3-8B and DeepSeek-VL2 Small-16B across 18 benchmarks. These include grounding and OCR tasks: TextVQA^{val} (Singh et al., 2019), RefCOCO, RefCOCO+ (Kazemzadeh et al., 2014), and RefCOCOg (Mao et al., 2016); comprehensive VQA benchmarks: Seed2-Plus (Li et al., 2024a), AI2D (Kembhavi et al., 2016), MMStar (Chen et al., 2024b), RealWorldQA (x.ai., 2024), SQA^{IMG} (Lu et al., 2022), VizWiz (Chen et al., 2022), MMVet (Yu et al., 2023), POPE (Li et al., 2023), MME, MMBench$^{\text{EN/CN}}$ (MMB$^{\text{EN/CN}}$) (Liu et al., 2024b), and GQA (Hudson and Manning, 2019); and video benchmarks: MVBench (Li et al., 2024b) and MLVU (Zhou et al., 2024). More details of benchmarks are in Appendix C. We compare HAP with cross-modal attention methods (e.g., FastV (Chen et al., 2024a), SparseVLM (Zhang et al., 2024b), PDrop (Xing et al., 2024), AutoPrune (Wang et al.)) and visual attention methods (e.g., VisionZip (Yang et al., 2025), VisPruner (Zhang et al., 2025b), CDPruner (Zhang et al., 2026), IVC-Prune (Sun et al., 2026b)). For fair comparison, we unify the token budget and compare FLOPs, KV cache, and inference speed. All experiments are conducted on a single NVIDIA RTX 3090.

4.2 Main Results

In Tab. 1, we evaluate HAP on LLaVA-1.5-7B across nine image understanding tasks. When reducing visual tokens from 576 to 32, HAP incurs only a 0.9% overall performance drop and 0.89 TFLOPs, substantially lower than competitors. Methods relying directly on cross-modal attention (*e.g.*, FastV, SparseVLM, and PDrop) suffer degradation exceeding 15%. Unimodal approaches such as VisionZip and VisPruner, lacking textual guidance, lose task-relevant regions at extreme compression ratios, incurring over 6% performance drop. AutoPrune employs mutual information for layer-

Table 1: Evaluation of our method on the LLaVA-1.5-7B model across nine datasets under three visual token compression levels (192, 128, and 64). The vanilla configuration uses 576 tokens and average 3.82T FLOPs. Relative score is the average ratio between the score and original score across all benchmarks.

Method	Venue	SQA	GQA	MMB^{EN}	MMB^{CN}	POPE	Text	MME	VizWiz	MMVet	Rel. Score	FLOPs
Original	-	69.5	61.9	64.6	58.1	86.1	58.2	1864	50.0	30.9	**100.0%**	3.82T
Retain 128 Tokens (22% Retention)												
FastV	ECCV'24	68.6	49.6	56.1	56.4	53.4	50.5	1490	51.3	26.3	86.6%	1.64T
SparseVLM	ICML'25	67.1	56.0	60.0	51.1	80.5	54.9	1696	51.4	29.6	93.9%	1.66T
PDrop	CVPR'25	**69.9**	56.0	61.1	56.6	82.3	55.1	1664	51.0	30.8	96.0%	1.53T
VisionZip	CVPR'25	68.9	57.6	62.0	53.4	83.2	56.8	1763	51.2	30.2	96.6%	1.44T
VisPruner	ICCV'25	68.2	57.4	62.7	57.3	84.6	57.0	1789	51.7	30.7	97.9%	1.45T
AutoPrune	ICCV'25	69.2	59.5	64.3	57.0	84.5	57.1	1785	49.1	30.3	98.0%	1.48T
HAP	-	69.7	**62.1**	**64.6**	**59.0**	**86.4**	**58.5**	**1872**	**51.8**	**33.4**	**102%**	1.42T
Retain 64 Tokens (11% Retention)												
FastV	ECCV'24	68.7	46.1	47.2	52.7	38.2	47.8	1255	50.8	19.6	77.3%	1.24T
SparseVLM	ICML'25	62.2	52.7	56.2	46.1	75.1	51.8	1505	50.1	23.3	86.0%	1.24T
PDrop	CVPR'25	**69.2**	43.9	53.3	50.5	75.9	48.9	1561	50.7	**30.7**	90.4%	1.17T
VisionZip	CVPR'25	69.0	55.1	60.1	51.9	77.0	55.5	1690	51.2	29.6	93.8%	1.12T
VisPruner	ICCV'25	69.1	55.4	61.3	55.1	80.4	55.8	1706	53.3	32.3	96.7%	1.14T
AutoPrune	ICCV'25	68.9	57.1	63.4	56.2	83.3	56.9	1745	49.4	29.9	96.9%	1.15T
HAP	-	69.3	**60.3**	**64.5**	**59.6**	**86.0**	**58.3**	**1883**	**52.1**	30.2	**100.3%**	1.12T
Retain 32 Tokens (5.6% Retention)												
FastV	ECCV'24	42.6	41.5	37.8	33.2	32.5	42.5	1184	51.7	20.7	65.4%	1.02T
SparseVLM	ICML'25	57.3	48.3	51.4	40.6	67.9	46.1	1347	51.9	18.6	78.3%	1.10T
PDrop	CVPR'25	**67.5**	40.3	51.4	48.9	68.6	45.7	1492	48.5	**25.1**	82.6%	0.98T
VisionZip	CVPR'25	67.2	51.5	57.3	49.1	74.3	50.9	1486	48.7	26.8	93.8%	0.91T
VisPruner	ICCV'25	69.2	52.2	58.4	52.7	72.7	53.9	1691	53.0	28.8	92.4%	0.93T
AutoPrune	ICCV'25	65.7	58.5	61.0	54.9	81.4	55.0	1732	49.7	29.2	94.9%	1.04T
HAP	-	69.2	**60.0**	**64.3**	**58.9**	**85.5**	**57.5**	**1875**	**51.7**	28.6	**99.1%**	0.89T

Table 2: Performance comparison on InternVL3 and Qwen2.5-VL VLMs with 25% visual tokens retained.

Model	Method	Venue	MMB^{EN}	MME	POPE	MVBench	AI2D	Seed2	RWQA	Relative Score(%)	FLOPs(T)
InternVL3-8B	original	-	83.4	2415	91.1	75.4	85.2	69.7	71.6	100%	8.15
	FastV	ECCV'24	80.4	2235	87.6	72.9	82.2	62.5	62.6	91.5% (↓8.5%)	3.05
	PDrop	CVPR'25	82.1	2298	88.2	73.4	83.1	63.8	63.8	94.2% (↓5.8%)	3.02
	VisPruner	ICCV'25	81.3	2372	89.4	72.1	82.2	64.9	66.1	95.3% (↓4.7%)	2.98
	HAP	-	**83.8**	**2417**	**90.6**	**74.2**	**83.7**	**69.1**	**71.9**	**99.5% (↓0.5%)**	2.90
Model	**Method**	**Venue**	MMB^{EN}	**MME**	**POPE**	MMB^{CN}	**AI2D**	**MLVU**	**RWQA**	**Relative Score(%)**	**FLOPs(T)**
Qwen2.5-VL-7B	original	-	83.5	2305	86.2	83.4	81.1	75.0	68.1	100%	6.10
	CDPruner	NurIPS'25	64.2	2200	72.5	79.0	56.3	63.1	57.3	84.1% (↓15.9%)	2.60
	IVC-Prune	ICLR'26	79.1	2183	81.6	79.0	76.8	71.0	64.5	94.7% (↓5.3%)	2.58
	HAP	-	**83.3**	**2293**	**85.9**	**82.7**	**80.3**	**74.3**	**68.6**	**99.1% (↓0.9%)**	2.55

wise budget allocation, neglecting the compression term essential for assessing cross-modal alignment quality and thus failing to identify reliable pruning signals with 5% performance drop. Notably, HAP improves results on MME, VizWiz, and MMB^{CN} when retaining 32 tokens, indicating accurate removal of hallucination-inducing visual tokens.

In Tab. 2, HAP achieves 99.5% and 99.1% relative scores on InternVL3-8B and Qwen2.5-VL-7B at 25% token retention, outperforming FastV, PDrop, and VisPruner (all >4% drops). On the challenging REC task (Tab. 3), HAP attains **81.5%** on Qwen3-VL-8B under 75% pruning, surpassing FastV and PDrop by over 30% absolute margins. Tab. 4 further shows HAP delivers a **1.96**× speedup on InternVL3-8B with only 34% KV cache overhead and 99.7% accuracy. Tab. 7 in Appendix D further reports that HAP attains a 103.0% relative average on DeepSeek-VL2 Small-16B (Wu et al., 2024) with 50% token retention, outperforming the baselines across all six general VQA benchmarks.

4.3 Ablation Study

Ablation of HAP. Fig. 4 shows that the gains stem from PAQ's joint formulation rather than heuristics. (a) PAQ (69.3%) substantially outperforms random pruning (64.0%), max/average attention (65.5% / 66.7%), and isolated criteria such as entropy-only (66.2%) and MI-only (67.2%). Gradient-based saliency (67.0%, computed via

Table 3: **Performance comparisons on Qwen-3-VL-8B** across 3 grounding benchmarks under 75% pruning ratio.

Method	Venue	RefCOCO			RefCOCO+			RefCOCOg		Average
		val	testA	testB	val	testA	testB	val	test	
Qwen3-VL-8B	-	91.58	93.27	86.76	85.80	90.22	79.78	88.76	88.24	100.0%
FastV	ECCV'24	43.57	46.81	40.86	39.47	43.78	36.02	43.04	42.69	47.7%
PDrop	CVPR'25	46.46	53.83	37.23	42.29	47.76	32.81	45.32	44.91	49.8%
HAP	-	**76.02**	**80.65**	**70.02**	**68.72**	**75.62**	**60.65**	**70.82**	**71.23**	**81.5%**

Table 4: Efficiency with InternVL3-8B at 25% tokens.

Method	KV Cache (%)	Perf. (%)	Speedup (×)
FastV	52	92.5	1.54
VisPruner	37	95.8	1.87
HAP (Ours)	**34**	**99.7**	**1.96**

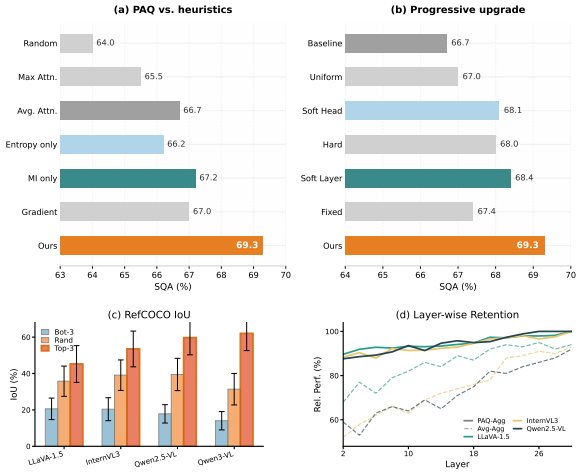

Figure 4: (a) PAQ *vs.* heuristic criteria on SQA. (b) Progressive integration of HAP components. (c) RefCOCO IoU of tokens selected by PAQ ranking across four VLMs. (d) Layer-wise retention stability (25% tokens): PAQ aggregation (solid) maintains robust performance across depths, while uniform aggregation (dashed) suffers from severe early-layer degradation caused by sink/dispersed heads.

backpropagated gradients) further verifies that PAQ achieves superior alignment without backpropagation. (b) Progressive integration yields monotonic improvements: soft head aggregation raises the baseline to 68.1%, soft layer fusion reaches 68.4%, and the full framework attains 69.3%. Replacing soft PAQ-weighted aggregation with uniform weighting, hard top-k selection, or a fixed per-group budget degrades results, confirming the principled design. (c) RefCOCO IoU validates that PAQ identifies spatially critical tokens: top-PAQ candidates consistently achieve higher IoU than random or bottom-PAQ baselines across all four architectures, with the gap widening on stronger models. (d) PAQ mitigates layer-wise heterogeneity: when retaining only 25% of visual tokens at each layer independently, PAQ-guided aggregation (solid curves) maintains stable relative performance (>85%) across all depths, whereas uniform aggregation (dashed curves) exhibits severe fluctuations—particularly in early layers where sink and dispersed heads dominate (e.g., LLaVA-1.5 drops below 55% at Layer 4). This confirms that PAQ reliably selects optimal attention patterns regardless of layer depth, eliminating the performance volatility that plagues fixed-layer pruning strategies (Fig. 2a).

Additionally, we evaluate HAP's robustness to prompt variations. We test three prompt styles: *standard* (original benchmark prompt), *detailed* (augmented with task-relevant image descriptions, e.g., scene context and object-relationship hints to encourage visual grounding), and *corrupt* (adversarially perturbed by random word substitution and key-token deletion to inject semantic noise). As shown in Table 5, all scores are reported as relative performance (%) against the original prompt. Despite the detailed prompt providing helpful visual context and the corrupt prompt introducing deliberate lexical interference, the maximum variation stays within 1.2%, confirming that HAP's pruning decisions are driven by intrinsic cross-modal alignment rather than superficial prompt artifacts.

5 Visualization Analysis.

Fig. 5 validates HAP on four VLMs across scene reasoning, counting, OCR, and digit recognition. After PAQ-weighted head aggregation, attention maps sharpen from noisy mixtures to concentrated regions (e.g., jersey numbers and pedestrian silhouettes), confirming that optimal heads dominate while sink and dispersed patterns are suppressed.

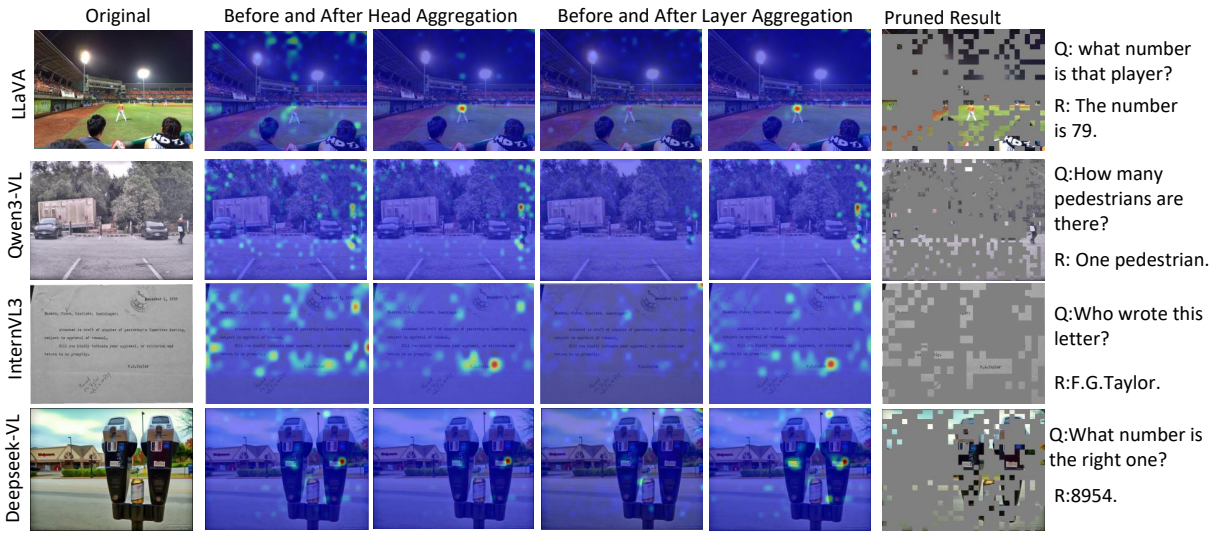

Figure 5: Qualitative visualization of HAP across LLaVA, Qwen3-VL, InternVL3, and DeepSeek-VL. Columns from left: original image; attention maps before/after PAQ-weighted head aggregation and before/after PAQ-weighted layer aggregation; final pruned token masks; and corresponding QA pairs.

Table 5: Prompt robustness of HAP (LLaVA-1.5-7B, 128 tokens retained). All scores are relative performance (%) against the original prompt. Δ denotes the maximum relative variation across prompt styles.

Benchmark	**Standard**	**Detailed**	**Corrupt**	**Δ (%)**
SQA	100.3	100.7	99.3	1.0
GQA	100.3	100.8	99.2	1.1
POPE	100.3	100.6	99.4	0.9
MME	100.4	100.9	99.3	1.2
Average	**100.3**	**100.8**	**99.3**	**1.0**

Table 6: Performance, latency, and KV cache usage analysis.

Retained tokens	576	128	64	32
Performance (%)	100	102	99.7	99.1
KV Cache (%)	100	36	25	20
Latency (s)	0.48	0.22	0.19	0.17
Speedup ($\times$)	1.00	2.18	2.53	2.82

Layer aggregation further refines these maps into compact, semantically coherent regions such as signature blocks and digit strings. The final pruned masks retain only task-critical tokens, enabling correct answers despite aggressive compression and verifying that HAP's two-level aggregation generalizes robustly across architectures.

Efficiency Analysis. In Tab. 6, we evaluate the practical efficiency of HAP on a single NVIDIA RTX 3090 (24GB). As visual tokens are progressively compressed, both latency and KV cache usage decrease substantially. Reducing tokens from 576 to 128 decreases latency from 0.48 s to 0.22 s, yielding a **2.18**$\times$ speedup while KV cache drops to 36%. Further compression to 64 tokens achieves **2.53**$\times$ speedup (0.19 s) with only 25% KV cache usage, retaining 99.7% performance. At 32 tokens, latency reduces to 0.17 s (**2.82**$\times$ speedup) with 20% KV cache, maintaining 99.1% accuracy. These results demonstrate that HAP effectively balances computational efficiency and model fidelity, substantially reducing memory and runtime demands with minimal performance degradation.

6 Conclusion

This paper presents HAP, a training free framework for visual token pruning in Vision Language Models. We identify heterogeneous cross-modal alignment across heads and layers, and propose the Prompt Grounded Attention Quality (PAQ) score to unify reliability assessment at both levels. HAP dynamically fuses attention heads and layer consensus via PAQ guided soft aggregation, allocating pruning budgets by alignment quality. Experiments across five architectures and 18 benchmarks demonstrate state of the art performance efficiency trade offs without hyperparameter tuning.

Acknowledgements

Please insert your acknowledgments here.

7 Limitations

HAP leaves several natural directions for extension. First, our framework operates during prefill

and leaves the visual KV cache untouched during decoding. Extending token consolidation to the autoregressive stage could further reduce memory for long outputs. Second, PAQ is derived from cross-modal attention, which assumes paired text queries. Applying similar head-adaptive criteria to unimodal compression remains future work. Finally, our evaluation focuses on static-image and short-video benchmarks. Head-level alignment patterns in streaming or ultra-long multimodal contexts warrant additional study.

Additionally, we acknowledge the use of AI-assisted writing tools (e.g., Claude) for language polishing, grammar correction, and stylistic refinement during manuscript preparation. These tools were not used to generate core technical contents such as research ideas, experimental results.

References

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. 2024. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13817–13827.

Chongyan Chen, Samreen Anjum, and Danna Gurari. 2022. Grounding answers for visual questions asked by visually impaired people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19098–19107.

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

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024b. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024c. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198.

Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 235–251. Springer.

Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. 2024a. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, and 1 others. 2024b. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206.

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

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

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

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024b. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Dongchen Lu, Yuyao Sun, Zilu Zhang, Leping Huang, Jianliang Zeng, Mao Shu, and Huo Cao. 2025. Internvl-x: Advancing and accelerating internvl series with efficient visual token compression. *arXiv preprint arXiv:2503.21307*.

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

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20.

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

Yuanhao Sun, Huawei Ji, Jiaxin Ding, Luoyi Fu, and Xinbing Wang. 2026a. Encore: Entropy-guided cropping and attention regularization for robust vision–language understanding. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12727–12731. IEEE.

Zhichao Sun, Yidong Ma, Gang Liu, Yibo Chen, Xu Tang, Yao Hu, and Yongchao Xu. 2026b. Ivc-prune: Revealing the implicit visual coordinates in lvlms for vision token pruning. *arXiv preprint arXiv:2602.03060*.

Henri Theil. 1970. On the estimation of relationships involving qualitative variables. *American Journal of Sociology*, 76(1):103–154.

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

Hanshi Wang, Zekun Xu, Jin Gao, Yufan Liu, Weiming Hu, Ke Wang, Zhipeng Zhang, and 1 others. Each complexity deserves a pruning policy. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zhenhailong Wang, Senthil Purushwalkam, Caiming Xiong, Silvio Savarese, Heng Ji, and Ran Xu. 2025. Dymu: Dynamic merging and virtual unmerging for efficient vlms. *arXiv preprint arXiv:2504.17040*.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, and 1 others. 2024. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*.

x.ai. 2024. Grok 1.5v: The future of ai models. Technical report.

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

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

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

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

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

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. 2026. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. *Advances in Neural Information Processing Systems*, 38:25438–25468.

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

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, and 1 others. 2024a. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257*.

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

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2024. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2(5):6.

A Derivation of the Closed-Form FLOPs

HAP partitions the L transformer layers into m stages of equal depth $\Delta l = L/m$. The stage index at layer l is $k = ml/L$, and the token count follows $N_v(l) = V/(1+k)^2$.

The total pruned FLOPs is obtained by summing Eq. 9 over all layers:

$$\mathrm{F}_{\text{pruned}}(m) = \sum_{l=1}^{L} \mathrm{F}\left(\frac{V}{(1+ml/L)^2}\right). \quad (11)$$

Because the layer index appears only through the stage ratio $k = ml/L$, the summation converges to a closed-form integral over the m stages. Let $u = 1 + ml/L$; then $\mathrm{d}l = \frac{L}{m}\mathrm{d}u$, and the integral spans exactly m stages from $u = 1$ to $u = 1+m$:

$$\mathrm{F}_{\text{pruned}}(m) \approx \frac{L}{m}\int_{1}^{1+m}\left[\frac{A}{u^2} + \frac{B}{u^4}\right]\mathrm{d}u, \quad (12)$$

where $A = 4VD^2 + 3VDd_{\text{ffn}}$ and $B = 2V^2D$.

Evaluating the two terms:

$$\int_{1}^{1+m}\frac{A}{u^2}\,\mathrm{d}u = A\left(1 - \frac{1}{1+m}\right), \quad (13)$$

$$\int_{1}^{1+m}\frac{B}{u^4}\,\mathrm{d}u = \frac{B}{3}\left(1 - \frac{1}{(1+m)^3}\right). \quad (14)$$

Combining these yields Eq. 10:

$$\mathrm{F}_{\text{pruned}}(m) \approx \frac{L}{m}\left[A\left(1 - \frac{1}{1+m}\right) + \frac{B}{3}\left(1 - \frac{1}{(1+m)^3}\right)\right] \quad (15)$$

The factor L/m reflects the per-stage layer depth, confirming that the total cost scales inversely with the stage count m for a fixed model depth L.

PAQ and pruning complexity. The per-head PAQ score (Eq. 3) is computed from the normalized cross-modal attention matrix $\mathbf{A}_h^{(l)} \in \mathbb{R}^{N_t \times N_v(l)}$. For each head, constructing the joint distribution $p(v_i, t_j) = \alpha_{ji}/N_t$ and the marginal $p(v_i) = \sum_j p(v_i, t_j)$ requires $\mathcal{O}(N_t N_v(l))$ operations. The mutual information $I(\mathbf{V};\mathbf{T})$ and entropy $H(\mathbf{V})$ each involve a single pass over the $N_t \times N_v(l)$ probability mass, adding another $\mathcal{O}(N_t N_v(l))$. With H attention heads, the total PAQ computation per layer is $\mathcal{O}(HN_tN_v(l))$.

Subsequent head aggregation (Eq. 5–7) computes H PAQ scores and applies softmax weighting, which is $\mathcal{O}(H)$ and thus absorbed into the per-layer constant. The token selection step (Eq. 8) averages the aggregated attention map over N_t text positions ($\mathcal{O}(N_tN_v(l))$) and sorts the $N_v(l)$ visual tokens by saliency ($\mathcal{O}(N_v(l)\log N_v(l))$). Because $N_t \gg \log N_v(l)$ in practice, the sorting cost is dominated by the attention aggregation.

Summing over all L layers and substituting $N_v(l) = V/(1+ml/L)^2$:

$$\sum_{l=1}^{L}\mathcal{O}(HN_tN_v(l)) = \mathcal{O}\left(HN_tV\sum_{l=1}^{L}\frac{1}{(1+ml/L)^2}\right). \quad (16)$$

Approximating the sum by an integral with $u = 1 + ml/L$:

$$\begin{aligned}\sum_{l=1}^{L}\frac{1}{(1+ml/L)^2} &\approx \frac{L}{m}\int_{1}^{1+m}\frac{1}{u^2}\,\mathrm{d}u \\ &= \frac{L}{m}\left(1 - \frac{1}{1+m}\right) \\ &<< \frac{L}{m}.\end{aligned} \quad (17)$$

Hence the total PAQ-and-pruning overhead is $\mathcal{O}(HN_tVL/m)$, which is strictly smaller than the unpruned per-layer attention cost $\mathcal{O}(N_tV)$ summed over L layers. Relative to the full forward-pass FLOPs (Eq. 10), the ratio is $\mathcal{O}(HD/(Am))$ where $A = 4VD^2 + 3VDd_{\text{ffn}}$, confirming that the overhead is negligible for typical configurations ($H \ll D$ and $m \geq 1$).

B Compatibility with FlashAttention

Modern Vision-Language Models predominantly adopt FlashAttention (Dao et al., 2022) or its variants as the default attention backend to eliminate the $O(N^2)$ memory bottleneck of standard self-attention. However, conventional token pruning

methods that rely on per-token attention importance typically require materializing the full $|\mathcal{T}| \times |\mathcal{V}|$ cross-modal attention matrix, thereby forfeiting the memory benefits of FlashAttention. HAP circumvents this limitation by design: the PAQ scores (Eq. 4) depend solely on *aggregated* cross-modal statistics per layer, rather than on individual token-level attention weights.

Why full attention materialization is unnecessary. For each layer, PAQ reduces to a low-dimensional quantity that collapses the vision-token dimension. Concretely, let $\mathcal{T}$ and $\mathcal{V}$ denote the text and vision token sets. The required statistic takes the form of a reduction over vision tokens,

$$\alpha_\ell \;=\; \sum_{j\in\mathcal{V}} f\left(\sum_{i\in\mathcal{T}} \mathrm{softmax}_j\left(\frac{Q_i K_j^\top}{\sqrt{d_k}}\right)\right),$$

where $f(\cdot)$ is an aggregation function (e.g., $\max$, mean, or log-sum-exp) dictated by the PAQ formulation. Because the inner softmax normalizes over text queries and the outer sum collapses the vision keys, the result is a *scalar per layer* (or a small vector if multiple statistical moments are tracked). Consequently, HAP never stores the intermediate $|\mathcal{T}| \times |\mathcal{V}|$ attention map.

Dual-flash aggregation. FlashAttention partitions the sequence into SRAM-resident blocks (tiles) and computes attention via online softmax, maintaining running maxima m and exponential sums ℓ to ensure numerical stability without materializing the full score matrix (Dao et al., 2022). To obtain the cross-modal aggregates required by PAQ, we extend this tiling strategy with an auxiliary fused reduction.

During the standard FlashAttention inner loop, each text block B_t and vision block B_v produces on-chip scaled dot-product scores $S_B = Q_{B_t} K_{B_v}^\top / \sqrt{d_k}$. Rather than discarding these scores after updating the attention output, we accumulate block-wise partial statistics into per-layer running buffers. For example, if the PAQ instantiation requires the maximal text-to-vision response, we update a running vector $\tilde{m}$ via $\tilde{m} \leftarrow \max(\tilde{m}, \max_{j\in B_v} S_{ij})$ across all $i \in B_t$; for mean-based moments, we accumulate block sums and counts. These buffers reside in SRAM and are flushed to HBM only once per layer.

Formally, let V_B, Q_B, K_B, S_B denote the value, query, key, and scaled dot-product blocks covering the text-to-vision interaction. The extended operation computes block-level aggregates:

$$A_{t2v}^{(B)} = \mathrm{Reduce}_B\Bigg(\mathrm{Softmax}\left(\frac{Q_B K_B^\top}{\sqrt{d_k}} - \max(S_B)\right) V_B \Bigg), \tag{18}$$

where Reduce_B denotes the block-wise reduction (sum, max, or higher-order moment) fused into the FlashAttention kernel. The final per-layer cross-modal statistic is obtained by aggregating block results across the sequence partition:

$$A_{t2v} \;=\; \mathrm{Concat}\left(\bigcup\nolimits_B A_{t2v}^{(B)}\right). \tag{19}$$

Because the reduction is associative and commutative, it integrates seamlessly with FlashAttention's parallel block scheduling without additional synchronization barriers.

Memory and computational overhead. Standard FlashAttention introduces $O(1)$ extra memory relative to the input size. Our extension increases this by exactly $L \times C$ scalars per sample, where L is the number of layers and C is a small constant ($C \leq 4$ for tracking max, sum, and variance). In practice, for a 32-layer model, this amounts to merely a few kilobytes—negligible compared with the hundreds of megabytes required to materialize a single full-precision attention matrix. The auxiliary reductions are fused into existing warp-level GPU operations, yielding a measured wall-clock overhead of $< 2\%$ relative to base FlashAttention.

Therefore, HAP is fully compatible with FlashAttention and other optimized attention implementations, inheriting their memory efficiency and hardware affinity without architectural modifications to the VLM backbone.

C Dataset

1. **GQA** (Hudson and Manning, 2019). The GQA consists of three components: scene graphs, questions, and images. The image segment encompasses images, along with the spatial features of images and the features of all objects within them. The questions in GQA are intended to assess the comprehension of visual scenes and the capacity to reason about diverse aspects of an image.

2. **MMBench** (Liu et al., 2024b).The MMBench benchmark thoroughly assesses a model's overall performance across multiple dimensions. It incorporates three tiers of ability dimensions. The first tier (L-1) comprises two core abilities: perception and reasoning. The second tier (L-2) builds upon the first, including six sub-abilities. The third tier (L-3) further refines the second, encompassing 20 specific ability dimensions. This layered framework enables a granular and comprehensive evaluation of the model's various capabilities.

3. **MME** (Zhang et al., 2024a). The MME benchmark is also a comprehensive benchmark carefully crafted to thoroughly evaluate various facets of a model's performance. It consists of 14 subtasks that specifically aim to evaluate both the model's perceptual and cognitive abilities. By employing manually constructed instruction-answer pairs and concise instruction design, it effectively alleviates problems such as data leakage and unfair evaluation of model performance.

4. **POPE** (Li et al., 2023). The POPE benchmark is mainly utilized to assess the extent of Object Hallucination in models. It redefines hallucination assessment by mandating the model to answer a series of specific binary questions concerning the presence of objects in images. Accuracy, Recall, Precision, and F1 Score are effectively used as reliable evaluation metrics to accurately gauge the model's hallucination level under three different sampling strategies.

5. **ScienceQA** (Lu et al., 2022). The ScienceQA benchmark encompasses a wide variety of fields, including natural science, language science, and social science. Within each subject, questions are categorized first by the topic, then by the category, and finally by the skill. This hierarchical classification leads to 26 topics, 127 categories, and 379 skills, providing a comprehensive and diverse range of scientific questions. It offers a thorough assessment of a model's capacities in multimodal understanding, multi-step reasoning, and interpretability.

6. **TextVQA** (Singh et al., 2019). The TextVQA benchmark centers on the thorough integration of various textual information embedded in images. It carefully assesses the model's text comprehension and reasoning abilities through a series of visual question-answering (QA) tasks featuring abundant textual data. Models are required to not only understand the visual content of the images but also read and reason about the text within the images to answer the questions accurately.

7. **MMVet** (Yu et al., 2023). The MMVet benchmark is developed on the premise that the fascinating capacity to tackle complex tasks is often achieved by a generalist model being able to combine various core vision-language competencies. MMVet defines 6 core VL capabilities and examines the 16 integrations of interest derived from the capability combination.

8. **VizWiz** (Chen et al., 2022). The VizWiz benchmark is primarily designed to address visual question-answering scenarios for visually impaired individuals. It consists of images from real-world, unconstrained scenes and corresponding questions that reflect the information needs of visually impaired people. This benchmark evaluates a model's ability to understand complex, real-world visual content and provide accurate, helpful answers, with a focus on supporting practical accessibility use cases.

9. **RealWorldQA** (x.ai., 2024). The RealWorldQA benchmark focuses on evaluating a model's capacity to answer questions about real-world scenarios, integrating multimodal information (including images and text) across diverse domains. It features questions rooted in everyday life, science, and society, aiming to test the model's ability to reason about real-world contexts, synthesize information from multiple sources, and provide coherent, factually accurate responses.

10. **MMStar** (Chen et al., 2024b). The MMStar benchmark is a multimodal benchmark tailored to evaluate models' performance in celebrity-related tasks. It comprises images of celebrities, textual descriptions, and questions spanning celebrity identification, career background, and entertainment context. This benchmark assesses a model's fine-grained visual recognition ability, multimodal infor-

mation integration, and knowledge reasoning in the entertainment domain, ensuring comprehensive evaluation of its competency in handling celebrity-centric multimodal data.

11. **AI2D** (Kembhavi et al., 2016). The AI2D benchmark is specifically tailored for diagram understanding and reasoning tasks. It comprises a large collection of scientific diagrams (including biology, physics, and chemistry illustrations) paired with domain-specific questions. This benchmark focuses on evaluating a model's ability to parse the structural components of diagrams (e.g., labels, arrows, symbols), grasp the logical relationships between elements, and perform accurate reasoning based on visual and scientific context—addressing the unique challenges of understanding non-photorealistic, information-dense visualizations.

12. **Seed2-Plus** (Li et al., 2024a). As an upgraded version of the Seed series benchmarks, Seed2-Plus is a comprehensive multimodal dataset designed to assess models' advanced capabilities in cross-modal understanding and complex reasoning. It integrates diverse data types such as images, text, and structured knowledge, covering multiple domains including daily life, science, and technology. The benchmark features questions that require multi-step logical inference, cross-modal information fusion, and factual verification, aiming to thoroughly test a model's robustness in handling complex multimodal scenarios and its ability to generate accurate, interpretable responses.

13. **RefCOCO** (Kazemzadeh et al., 2014). The RefCOCO dataset is constructed from MS-COCO images to evaluate visual grounding performance. It contains short referring expressions (3.6 words on average) collected via an interactive game interface, encompassing 142,210 expressions across 19,994 images. The dataset is split into train, validation, testA (focused on people), and testB (focused on objects) subsets, with accuracy defined by IoU > 0.5 between predicted and ground-truth bounding boxes.

14. **RefCOCO+** (Kazemzadeh et al., 2014). The RefCOCO+ dataset shares the same image source and collection paradigm as RefCOCO but imposes a stricter constraint: referring expressions must not contain absolute location words (e.g., "left", "right"). This requires models to rely purely on appearance and relational descriptions rather than spatial cues, resulting in 141,564 expressions that test robust visual-linguistic grounding without positional shortcuts.

15. **RefCOCOg** (Mao et al., 2016). The RefCOCOg dataset distinguishes itself from RefCOCO and RefCOCO+ through longer, more complex referring expressions (8.4 words on average) collected in non-interactive settings. With 95,010 expressions over 25,799 images, it features fewer same-type objects per query (1.6 on average), demanding finer-grained discrimination and more comprehensive language understanding for accurate object localization.

16. **MVBench** (Li et al., 2024b). The MVBench is a comprehensive multi-modal video understanding benchmark covering 20 challenging tasks that cannot be solved with single-frame analysis. It spans temporal skills ranging from perception to cognition, including action understanding, object interaction, scene transition, and counterfactual inference. By converting public video annotations into multiple-choice questions via a static-to-dynamic paradigm, it ensures fair evaluation grounded in video annotations rather than biased LLM scoring.

17. **MLVU** (Zhou et al., 2024). The MLVU is the first comprehensive benchmark designed for multi-task long video understanding, featuring videos ranging from 3 minutes to 2 hours in duration. It encompasses nine distinct evaluation tasks, including topic reasoning, anomaly recognition, needle QA, ego reasoning, plot QA, action ordering, and action counting. This benchmark rigorously tests MLLMs' capabilities in leveraging both global and local temporal information across extended video contexts.

D DeepSeek-VL2 Small-16B.

Table 7 supplements the main results by reporting HAP on DeepSeek-VL2 Small-16B (Wu et al., 2024) across six general VQA benchmarks: SEED,

Table 7: Performance on DeepSeek-VL2 Small-16B across six benchmarks.

Method	Token↓	SEED	MMB	MMS	RWQA	HallB	AI2D	Rel. Avg.
Vanilla	100%	76.9	79.2	57.7	70.3	43.8	82.0	100%
FastV	54%	75.6	78.2	55.9	69.0	42.7	81.0	98.1%
PDrop	57%	76.8	79.1	57.3	69.7	44.5	81.8	99.9%
IVC-Prune	52%	77.0	79.3	57.7	70.3	44.3	81.8	100.2%
HAP	50%	79.4	81.6	59.4	72.4	45.1	84.4	103.0%

MMB, MMS, RWQA, HallB, and AI2D. HAP retains only 50% of the visual tokens yet achieves a 103.0% relative average, outperforming the unpruned Vanilla and other pruning methods on every benchmark in this group. This indicates that the proposed alignment-based aggregation not only preserves but also enhances model capabilities under aggressive compression.