HEAT: HIERARCHICAL ELASTIC ADAPTIVE TOKEN SPARSIFICATION VIA DEPTH SCALING

Anonymous authors
Paper under double-blind review

ABSTRACT

Excessively long visual token sequences introduce substantial computational and memory overhead during the prefill stage of large vision-language models (LVLMs). Existing training-free visual token reduction methods alleviate this burden by pruning visual tokens at predefined LLM layers. However, fixed-depth pruning implicitly assumes that the same reduction depths are appropriate for all inputs, resulting in a mismatch between pruning depth and visual processing demand. To address this issue, we propose **H**ierarchical **E**lastic **A**daptive **T**oken Sparsification (HEAT), a training-free framework that scales pruning depth according to the evolving relevance state of each input. Specifically, HEAT introduces a cross-layer relevance monitor to quantify visual relevance concentration under the current token-retention budget, facilitating input-specific identification of appropriate pruning depths. Building on this, HEAT further develops a depth-adaptive threshold scheduler to adjust the pruning criterion according to both reduction ratio and model depth, supporting progressive visual token reduction at appropriate depths. Experiments across five LVLM backbones and a broad range of image, document, and video benchmarks demonstrate the effectiveness and generalizability of HEAT. With 66.7% of visual tokens removed, HEAT preserves 99.4% of full-token performance, reduces visual KV-cache memory by 66%, and accelerates LLM prefill by up to $1.59\times$. Code is available at `https://anonymous.4open.science/r/HEAT-C08E/`.

1 INTRODUCTION

Large vision-language models (LVLMs) represent high-resolution images, documents, and videos as long sequences of visual tokens that participate in computation throughout the LLM backbone (Liu et al., 2023; Bai et al., 2025; Chen et al., 2024c). While dense visual tokens preserve fine-grained visual evidence, they substantially increase attention computation, prefill latency, and key–value (KV) cache memory. This overhead becomes particularly pronounced for document and video inputs, making high-resolution LVLM inference increasingly computationally and memory intensive.

Training-free visual token reduction (VTR) provides an effective means of reducing this cost without updating the pretrained model. Existing methods primarily determine *which* visual tokens should be retained according to attention, similarity or diversity (Chen et al., 2024a; Yang et al., 2025; Alvar et al., 2025; Zhang et al., 2025c;a; Wen et al., 2025b), while recent approaches further distribute token reduction across multiple layers or adapt the retained token ratio (Xing et al., 2024; Liu et al., 2024a; Ye et al., 2025b; Chen et al., 2026a). Despite these advances, the depth at which visual tokens are reduced is typically predefined or governed by a fixed multi-layer schedule, implicitly assigning similar visual processing depth to inputs with different processing demands.

More fundamentally, fixed-depth pruning assumes that all inputs require the same visual processing depth before token reduction. In practice, different inputs respond differently to pruning at different LLM layers, indicating that the required visual processing depth varies with the image and question. This motivates a central question: *should different visual inputs undergo token reduction at the same LLM depths?* Figure 1 provides representative examples. Under the same final token budget, fixed-depth methods fail to retain visual regions relevant to the question and produce incorrect answers, whereas adapting the reduction depths preserves these regions and yields correct predictions.

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

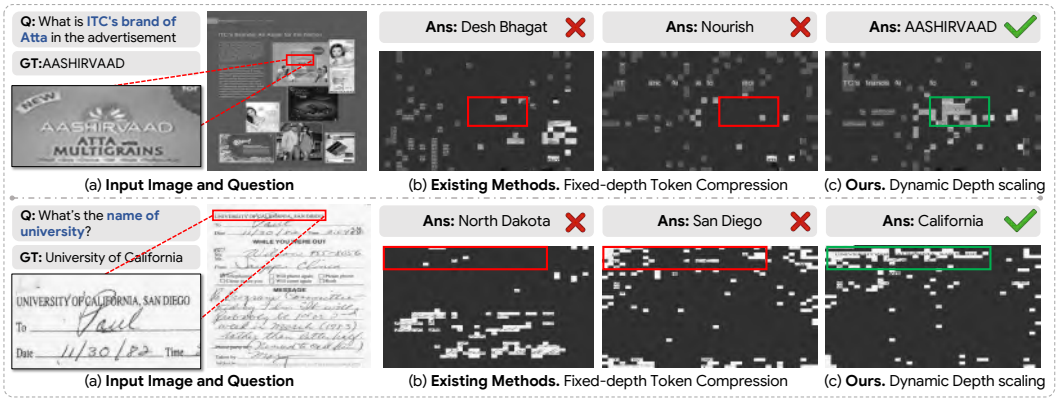


Figure 1: **Fixed-depth pruning versus adaptive depth scaling.** Retained visual regions and model predictions are compared under a matched final token budget.

The limitation is not that a particular pruning depth is universally ineffective, but that fixed pruning depths cannot accommodate different visual processing demands. A *shallow* depth may remove informative visual evidence before sufficient cross-modal processing has occurred, whereas an unnecessarily *late* depth retains redundant visual tokens through additional LLM blocks. Therefore, *input-adaptive* depth scaling is needed to adjust the execution depth of each reduction stage according to the evolving relevance state of each input.

To address this fixed-depth mismatch, we propose **HEAT**, a training-free framework for multi-stage adaptive depth scaling based on the evolving relevance state of each input, as illustrated in Figure 2. Its cross-layer relevance monitor (CRM) quantifies the text-conditioned relevance covered by the current stage budget, while its depth-adaptive threshold scheduler (DTS) uses reduction severity and model depth to trigger or defer each stage. This assigns input-specific depths to successive reduction stages under a common final token budget and requires no additional training. Decomposing the target ratio into progressive stages allows later decisions to use deeper cross-modal representations instead of relying on a single representation state.

In summary, our main contributions are as follows:

1. We discover that appropriate pruning depths vary across inputs and systematically characterize how this variation follows evolving text-conditioned visual relevance. To the best of our knowledge, we are the first to formulate this as multi-stage adaptive depth scaling for training-free VTR, revealing the fixed-depth mismatch in existing methods.
2. We propose HEAT, a training-free framework for hierarchical elastic adaptive token sparsification that converts evolving semantic relevance into input-specific depth-scaling decisions. Its cross-layer relevance monitor and depth-adaptive threshold scheduler determine the execution depth of each progressive reduction stage under a matched final token budget.
3. We validate HEAT across five LVLM backbones and diverse benchmarks. With 66.7% of visual tokens removed, HEAT preserves 99.4% of full-token performance, reduces visual KV-cache memory by 66%, and accelerates LLM prefill by up to $1.59\times$.

2 Related Work

Training-free visual token reduction (VTR) reduces the inference overhead of LVLMs without modifying or retraining the pretrained model. Existing methods primarily identify informative visual tokens according to similarity, clustering, or related structural cues (Yang et al., 2025; Alvar et al., 2025; Dhouib et al., 2025; Wen et al., 2025b; Ye et al., 2025a; Tong et al., 2025; Zhang et al., 2026). Other approaches further incorporate language-side signals, such as LLM attention, instruction relevance, or hierarchical interactions, to improve query-aware token selection (Chen et al., 2024a; Shang et al., 2024; Zhang et al., 2025a; Guo et al., 2025; Liu et al., 2026; Zou et al., 2025; Huang et al., 2026; Sun et al., 2025). These methods substantially improve *which* tokens are retained af-

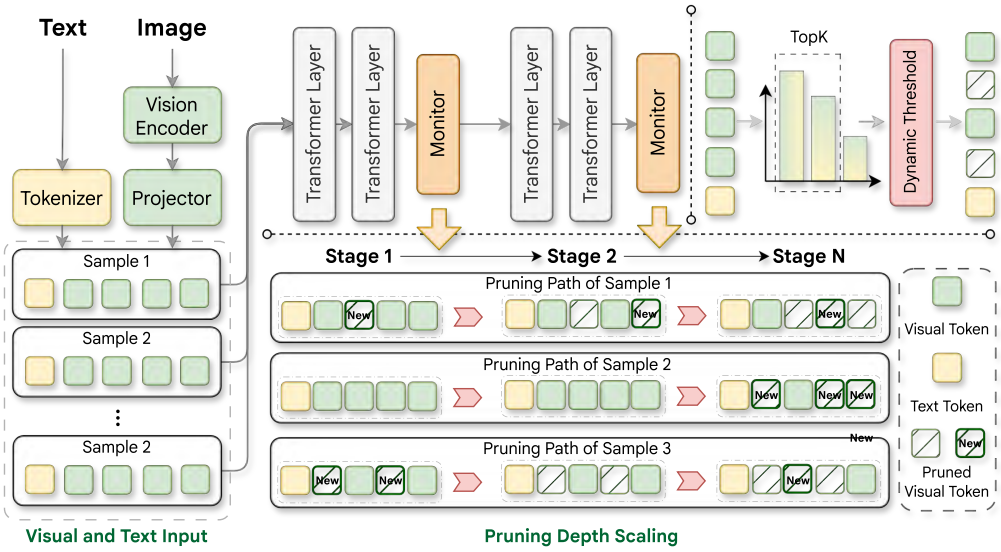


Figure 2: **Overview of HEAT.** For each input, the cross-layer relevance monitor measures the relevance mass covered by the fixed top-K budget of the pending stage. When the depth-dependent threshold is met, HEAT triggers a reduction stage and any unfinished stages are completed at the fallback layer. The three rows illustrate different input-specific depth-scaling patterns.

ter reduction. However, token selection is typically coupled with a predefined intervention depth, leaving the required amount of visual processing before pruning largely unmodeled.

Recent methods extend single-stage pruning by distributing token reduction across multiple layers, dynamically adjusting retention ratios, or exploiting temporal redundancy in long visual sequences (Xing et al., 2024; Liu et al., 2024a; Ye et al., 2025b; Chen et al., 2026a; Zhang et al., 2025b; Chen et al., 2026b; Huang et al., 2025; Tao et al., 2025; Zeng et al., 2026). SparseVLM similarly performs text-guided progressive sparsification and uses the rank of the language-to-vision attention matrix to adapt the number of visual tokens removed across decoder layers (Zhang et al., 2025c). These studies demonstrate the benefit of progressive or adaptive token allocation over a single fixed reduction step. Their adaptation mainly concerns how many tokens are removed at the available reduction layers, whereas HEAT maintains a common stage-wise budget and adapts the depth at which each pending stage is executed. This allows the timing of irreversible token reduction to follow the evolving cross-modal processing within the LLM.

Existing VTR methods have extensively explored *which* visual tokens should be retained and *how many* tokens should remain. While recent studies have begun to characterize layer-wise variations in visual attention and cross-modal relevance (Yin et al., 2025; Chen et al., 2026a; Zhang et al., 2025b; Chen et al., 2026b; Wen et al., 2025a), these signals are primarily used to improve token selection or retention allocation. Explicitly assigning a prescribed set of reduction stages to input-specific depths remains less explored. HEAT addresses this question by dynamically determining the execution depth of each stage from the evolving relevance state of the input.

3 Preliminary: Fixed-Depth Visual Token Reduction

VLM decoders model cross-modal interactions by jointly processing visual and text tokens through causal self-attention. At layer l, let N_l and N_t denote the numbers of visual and text tokens, respectively, and let d denote the hidden width. We write the hidden sequence as $\mathbf{H}_l = [\mathbf{V}_l; \mathbf{T}_l] \in \mathbb{R}^{(N_l+N_t)\times d}$, where $\mathbf{V}_l \in \mathbb{R}^{N_l\times d}$ and $\mathbf{T}_l \in \mathbb{R}^{N_t\times d}$ denote the visual token and text token representations. For the h-th attention head, attention-based visual token reduction methods typically first average the attention maps across heads:

$$\mathbf{A}_l^{(h)} = \mathrm{softmax}\Big(\mathbf{Q}_l^{(h)}(\mathbf{K}_l^{(h)})^\top/\sqrt{d_h}\Big), \quad \overline{\mathbf{A}}_l = \frac{1}{H}\sum_{h=1}^{H}\mathbf{A}_l^{(h)}. \tag{1}$$

They then derive a visual token importance score $\mathbf{s}_l \in \mathbb{R}^{N_l}$ from the relevant part of $\overline{\mathbf{A}}_l$, such as the visual-attention region or the text-to-vision attention region. For example, using text-to-vision attention gives $\mathbf{s}_l = \frac{1}{N_t} \sum_{i \in \mathcal{I}_t} \overline{\mathbf{A}}_l[i, \mathcal{I}_v]$, where $\mathcal{I}_t$ and $\mathcal{I}_v$ denote the index sets of text tokens and visual tokens. Given a target reduction ratio $r \in (0, 1)$, existing methods typically retain the visual tokens with the highest scores at a *predefined* layer l:

$$\widetilde{N}_l = \mathrm{round}\big((1-r)N_l\big), \qquad \mathcal{S}_l = \mathrm{TopK}\Big(\mathbf{s}_l, \widetilde{N}_l\Big). \tag{2}$$

Although different methods construct $\mathbf{s}_l$ from different attention regions, they largely share the same paradigm: *attention-derived scores at a predefined layer are used to rank and prune visual tokens*. This implicitly assumes that the attention at the selected layer is already sufficiently reliable for token selection. However, such reliability can vary across layers and inputs, making a fixed pruning depth potentially suboptimal.

4 Proposed Approach: HEAT

We propose HEAT, a training-free framework that adaptively determines the layer of each progressive visual-token reduction stage according to the evolving relevance state of each input. The cross-layer relevance monitor measures how much text-relevant visual information is covered by the budget of the next reduction stage, while the depth-adaptive threshold scheduler determines whether to execute that stage at the current layer or defer it to a deeper layer. These components reduce single-layer importance misestimation and adaptively allocate visual computation according to input-specific pruning sensitivity.

Progressive Budget Decomposition. Given a target reduction ratio $r \in (0, 1)$, HEAT decomposes the final budget into M equal multiplicative stages. Let $N^{(m)}$ denote the number of visual tokens after m stages have been completed, where $N^{(0)} = N_0$. We define the per-stage retention factor as

$$\gamma_r = (1-r)^{1/M}, \qquad \gamma_r^M = 1 - r. \tag{3}$$

While $m < M$ stages have been completed, the token budget of the next stage is

$$K^{(m+1)} = \mathrm{round}\Big(\gamma_r N^{(m)}\Big). \tag{4}$$

This budget remains unchanged while HEAT scans candidate layers for stage $m+1$. Only after the stage is executed do we set $N^{(m+1)} = K^{(m+1)}$ and compute the budget of the following stage. Thus, model depth controls where each stage is applied, not how many tokens that stage retains.

4.1 Cross-Layer Relevance Monitor (CRM)

The key challenge is to measure how much text-relevant visual information can be represented by the fixed budget of the pending stage. Rather than relying on a single-layer attention snapshot, CRM computes lightweight text–vision relevance at each candidate layer. At layer l, after m stages have been completed, let $\mathbf{v}_{l,i} \in \mathbb{R}^d$ and $\mathbf{t}_{l,t} \in \mathbb{R}^d$ denote the i-th visual token and t-th text token, respectively. We compute

$$z_{l,i} = \frac{1}{N_t} \sum_{t=1}^{N_t} \frac{\mathbf{v}_{l,i}^\top \mathbf{t}_{l,t}}{\|\mathbf{v}_{l,i}\|_2 \|\mathbf{t}_{l,t}\|_2}, \qquad p_{l,i} = \frac{\exp(z_{l,i}/T)}{\sum_{j=1}^{N^{(m)}} \exp(z_{l,j}/T)}, \tag{5}$$

where T is the softmax temperature. For the pending stage $m+1$, CRM forms a top-K monitoring set using the fixed budget $K^{(m+1)}$ and measures its cumulative relevance mass $C_l^{(m+1)}$:

$$\mathcal{G}_l^{(m+1)} = \mathrm{TopK}\Big(\mathbf{p}_l, K^{(m+1)}\Big), \qquad C_l^{(m+1)} = \sum_{i \in \mathcal{G}_l^{(m+1)}} p_{l,i}. \tag{6}$$

The set $\mathcal{G}_l^{(m+1)}$ is used only to monitor relevance coverage; the tokens retained after reduction are selected separately in Section 4.2. Since $K^{(m+1)}/N^{(m)} \approx \gamma_r$, a uniform relevance distribution gives $C_l^{(m+1)} \approx \gamma_r$. A larger value indicates that the same token budget covers a more concentrated share of text-relevant visual information, whereas a lower value indicates that this information is distributed across more visual tokens. CRM does not change the stage budget or estimate semantic redundancy; it measures relevance concentration under the prescribed budget.

4.2 Depth-Adaptive Threshold Scheduler (DTS)

The stage budget is fixed while candidate layers are scanned, but the required relevance mass varies with model depth. DTS controls this depth-dependent criterion without changing $K^{(m+1)}$.

Let L_s and L_f denote the first monitored layer and the fallback layer, respectively, and define the normalized depth as $u_l = (l - L_s)/(L_f - L_s) \in [0, 1]$. We interpolate the relative coverage margin using $\delta_r(l) = \delta_r^{\text{last}} + (\delta_r^{\text{start}} - \delta_r^{\text{last}})(1 - u_l^2)$, which yields the depth-adaptive threshold

$$\tau_r(l) = \gamma_r \left[1 + \delta_r^{\text{last}} + \left(\delta_r^{\text{start}} - \delta_r^{\text{last}}\right) \left(1 - \left(\frac{l - L_s}{L_f - L_s} \right)^2 \right) \right]. \tag{7}$$

Here, γ_r is the uniform-mass reference induced by the stage budget, and $\delta_r(l)$ specifies the required margin above that reference. At each monitored layer, HEAT executes stage $m + 1$ when

$$C_l^{(m+1)} \geq \tau_r(l). \tag{8}$$

If the condition is not met, no tokens are removed: $N^{(m)}$ and $K^{(m+1)}$ remain unchanged, and the same stage is evaluated at the next monitored layer. The margin is stricter at shallow layers, where premature pruning affects more subsequent blocks, and gradually decreases toward the fallback layer.

Token Selection. CRM and DTS determine *when* to reduce visual tokens, while the attention-based importance score determines *which* tokens to retain. Once stage $m + 1$ is activated at layer l, HEAT applies

$$\mathcal{S}_l^{(m+1)} = \text{TopK}\Big(\mathbf{s}_l, K^{(m+1)}\Big), \qquad \widetilde{\mathbf{V}}_l = \mathbf{V}_l[\text{sort}(\mathcal{S}_l^{(m+1)})], \tag{9}$$

where $\mathbf{s}_l$ is the visual token importance score defined in Section 3. The score is recomputed only at activated layers from the current query and key states, while the backbone continues to use FlashAttention for its standard forward pass. After selection, stage $m + 1$ is marked as complete and the next stage budget is computed from the retained token count.

Fallback and Budget Completion. All inputs share the same M conceptual stages, but several unfinished stages may be combined into one pruning operation at the fallback layer. If only $m < M$ stages have been completed before L_f, HEAT applies the same token selector once more and retains

$$K_f = \text{round}\left(N_f \gamma_r^{M-m}\right) = \text{round}\left(N_f (1 - r)^{(M-m)/M}\right), \tag{10}$$

where N_f is the number of visual tokens entering the fallback layer. No additional coverage test is performed at L_f. Consequently, inputs can execute different numbers of pruning operations: some stages are applied at separate monitored layers, while the remaining stages are merged at the fallback layer. Since $\gamma_r^M = 1 - r$, every input terminates with approximately $(1 - r)N_0$ visual tokens despite these differences in execution depth.

5 Experiments

Following the motivation in Section 1, we investigate three questions: **Q1**: Is HEAT effective and efficient compared with existing training-free VTR methods? **Q2**: Where do its performance gains come from? **Q3**: Does adaptive pruning better preserve query-relevant visual evidence than fixed-depth selection? We answer these questions through benchmark evaluation, controlled ablations, and qualitative analysis.

5.1 Main Results

To answer **Q1**, we compare HEAT with existing training-free VTR methods under matched final retention budgets across image, document, and video benchmarks. We further report LLM-prefill latency and memory consumption to evaluate practical efficiency.

❶**General-image understanding.** As shown in Table 1, HEAT consistently achieves the strongest overall performance across both backbones and retention budgets. At 33.3% retention, HEAT

Table 1: **General-image performance across Qwen2.5-VL-7B and InternVL2.5-8B.**

Method	MME	MMB(EN)	MMB(CN)	MMStar	POPE	RWQA	BLINK	AI2D	GQA	VQAv2
***Qwen2.5-VL-7B**, Upper Bound, All Tokens (100%)*										
Vanilla	2311.7	84.1	80.2	62.2	87.6	68.2	56.0	82.3	60.9	82.9
Average Retain 33.3% Tokens										
Random	2256.0	79.7	76.4	56.8	85.0	64.8	48.5	77.7	58.7	79.0
FastV (ECCV'24)	2192.0	79.8	75.9	52.2	79.2	58.8	53.5	76.3	54.5	74.9
PyramidDrop (CVPR'25)	2273.2	81.8	79.3	57.8	87.2	67.1	54.3	79.7	59.0	81.4
DivPrune (CVPR'25)	2282.5	82.0	78.2	57.5	86.8	65.2	45.3	80.3	59.1	80.5
DART (EMNLP'25)	2210.7	80.6	76.9	54.4	86.0	63.7	50.9	79.6	59.4	79.2
V^2Drop (CVPR'26)	2279.2	82.3	79.1	58.4	84.5	63.1	54.5	80.2	58.2	78.7
HEAT (Ours)	**2311.4**	**82.4**	**79.8**	**60.5**	**87.9**	**69.9**	**54.6**	**80.6**	**59.7**	**81.9**
Average Retain 22.2% Tokens										
Random	2199.3	77.5	74.5	53.9	83.8	62.6	46.9	75.2	57.3	76.6
FastV (ECCV'24)	2113.1	78.2	75.1	48.8	76.1	54.2	52.6	73.4	51.6	71.3
PyramidDrop (CVPR'25)	2247.4	80.5	77.2	55.4	86.4	67.5	53.7	77.9	57.0	80.2
DivPrune (CVPR'25)	2231.6	80.0	78.2	53.0	86.4	63.0	44.0	78.3	57.6	78.9
DART (EMNLP'25)	2131.6	79.6	78.5	50.6	83.5	63.1	49.4	76.2	56.9	76.1
V^2Drop (CVPR'26)	2282.5	80.2	77.0	56.4	83.6	61.6	53.4	78.1	57.1	76.8
HEAT (Ours)	**2282.9**	**81.7**	**78.7**	**58.9**	**87.7**	**69.4**	**54.3**	**79.6**	**58.8**	**81.1**
***InternVL2.5-8B**, Upper Bound, All Tokens (100%)*										
Vanilla	2322.1	84.9	83.0	61.0	90.7	68.4	55.6	83.3	63.2	81.2
Average Retain 33.3% Tokens										
Random	2177.0	79.1	78.4	54.7	89.9	61.3	49.1	77.6	60.4	77.4
FastV (ECCV'24)	2201.6	78.9	77.2	54.4	89.7	62.5	54.1	76.6	60.0	77.0
PyramidDrop (CVPR'25)	2314.4	84.5	82.4	60.4	90.5	68.0	54.8	81.8	62.5	80.8
DivPrune (CVPR'25)	2254.7	82.6	80.9	57.3	90.4	62.9	51.5	81.0	61.0	78.7
DART (EMNLP'25)	2282.0	83.2	80.8	56.0	90.2	67.2	48.5	80.0	61.6	80.0
V^2Drop (CVPR'26)	2299.1	83.8	81.4	58.2	90.1	67.3	55.0	81.7	59.6	76.2
HEAT (Ours)	**2326.6**	**84.7**	**82.6**	**60.5**	**90.6**	**68.6**	**55.1**	**82.1**	**63.0**	**81.0**
Average Retain 22.2% Tokens										
Random	2159.8	76.2	75.9	51.0	89.0	59.5	45.8	74.4	58.9	75.2
FastV (ECCV'24)	2133.3	74.1	72.4	48.1	88.6	60.8	52.3	73.4	58.0	73.0
PyramidDrop (CVPR'25)	2314.7	84.1	82.0	58.9	90.5	66.9	54.8	81.2	62.1	80.5
DivPrune (CVPR'25)	2150.5	81.2	79.6	55.2	90.4	61.3	50.9	78.8	60.4	77.4
DART (EMNLP'25)	2237.9	81.4	80.7	52.7	89.6	65.2	45.4	77.5	60.0	78.2
V^2Drop (CVPR'26)	2291.6	83.4	81.1	57.0	89.7	65.0	54.5	81.3	58.5	75.0
HEAT (Ours)	**2322.8**	**84.4**	**82.5**	**59.7**	**90.7**	**68.1**	**54.9**	**81.8**	**62.8**	**80.8**

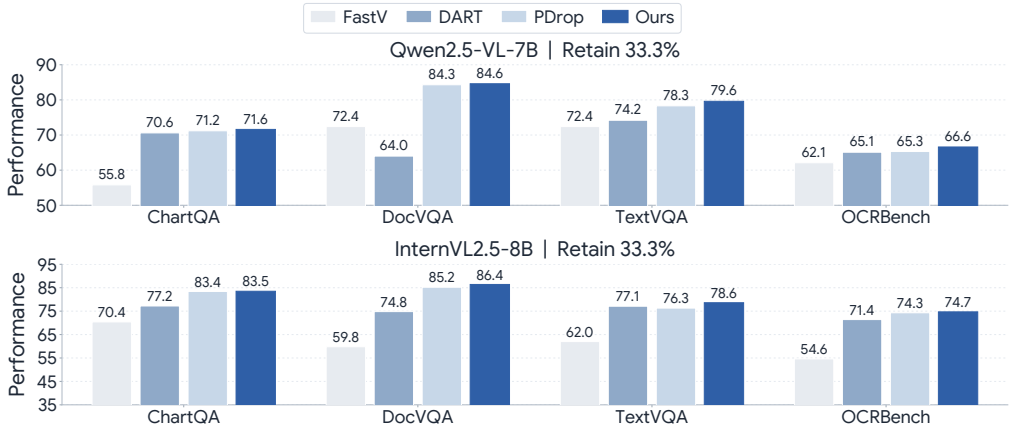


Figure 3: **Document-understanding performance across Qwen2.5-VL-7B and InternVL2.5-8B.**

reaches 60.5 MMStar on both Qwen2.5-VL-7B and InternVL2.5-8B. Even at 22.2% retention, it maintains 58.9/59.7 MMStar and 2282.9/2322.8 MME, respectively. These results show that input-adaptive depth scaling preserves visual information more reliably than applying the same fixed pruning depths to every input. Results under more aggressive budgets and on LLaVA are provided in Appendix 7 and Appendix B.4.

Table 2: **LLM-prefill efficiency at 66.7% visual-token reduction.** Measurements include throughput, prefill latency, speedup, FLOPs, KV-cache, peak GPU memory, and depth-scaling overhead.

Method	Throughput (KToken/s)↓	Prefill (ms)↓	Speedup Ratio↑	FLOPs (T)↓	KV Cache (GiB)↓	Peak GPU Memory (GiB)↓	Algorithm (ms)↓
Image / MME							
Vanilla	14.2	161.76	1.00×	33.5	0.123	15.99	–
V^2Drop (CVPR'26)	20.1	114.03	1.42×	19.9	0.042	15.89	**1.66**
HEAT (Ours)	**22.0**	**104.57**	**1.55×**	**19.7**	**0.042**	**15.89**	2.92
OCR / DocVQA							
Vanilla	13.7	175.09	1.00×	35.0	0.128	15.99	–
V^2Drop (CVPR'26)	20.1	118.92	1.47×	20.7	0.044	15.89	**1.66**
HEAT (Ours)	**21.7**	**110.45**	**1.59×**	**19.4**	**0.044**	**15.89**	2.87

Table 3: **Video-understanding performance across Qwen2.5-VL-7B and InternVL2.5-8B.**

Method	Qwen2.5-VL-7B					InternVL2.5-8B				
	VMME	VMME$^{w/o}$	MVBench	MLVU	Per. Test	VMME	VMME$^{w/o}$	MVBench	MLVU	Per. Test
Upper Bound, All Tokens (100%)										
Vanilla	61.1	64.4	68.2	59.5	70.5	63.9	66.2	72.8	65.6	66.5
Average Retain 33.3% Tokens										
Random	59.4	62.3	66.4	58.0	67.5	61.9	63.8	70.4	63.0	63.3
FastV (ECCV'24)	60.7	63.0	67.5	57.7	67.3	62.9	65.1	71.8	63.7	63.1
PyramidDrop (CVPR'25)	60.7	63.2	67.5	58.1	68.4	62.3	65.0	72.8	64.3	64.3
DivPrune (CVPR'25)	59.0	62.1	66.7	58.7	66.5	61.6	63.4	69.1	64.1	64.2
DART (EMNLP'25)	58.4	61.4	65.7	56.8	65.7	62.6	64.9	72.0	64.5	63.0
V^2Drop (CVPR'26)	59.9	63.3	65.8	58.6	68.3	62.5	64.7	70.2	57.7	64.5
HEAT (Ours)	**61.1**	**63.6**	**67.5**	**59.0**	**68.4**	**62.7**	**65.1**	**72.9**	**64.8**	**64.5**
Average Retain 22.2% Tokens										
Random	58.3	61.4	65.4	57.6	66.5	60.4	62.2	68.5	64.3	62.9
FastV (ECCV'24)	59.4	61.7	66.6	56.1	66.1	62.0	64.4	70.3	64.4	62.5
PyramidDrop (CVPR'25)	60.7	62.2	66.8	57.3	67.7	62.1	65.1	72.6	64.1	63.9
DivPrune (CVPR'25)	57.9	60.9	66.1	57.6	65.3	60.3	62.7	68.5	64.0	63.9
DART (EMNLP'25)	56.8	60.1	63.3	55.8	63.6	61.6	63.5	70.7	64.2	62.0
V^2Drop (CVPR'26)	60.3	62.4	64.6	58.0	67.7	62.1	64.6	69.7	55.9	63.8
HEAT (Ours)	**60.8**	**63.2**	**67.4**	**58.3**	**67.8**	**62.2**	**65.2**	**72.9**	**64.3**	**64.0**
Average Retain 11.1% Tokens										
Random	56.5	59.4	63.6	56.3	64.3	58.0	60.6	65.7	62.5	62.0
FastV (ECCV'24)	56.6	59.3	64.1	51.8	64.2	58.9	61.4	64.0	60.2	62.0
PyramidDrop (CVPR'25)	57.4	61.1	64.6	55.7	65.9	61.5	64.7	71.2	64.1	62.8
DivPrune (CVPR'25)	56.8	59.4	64.2	56.5	63.2	59.0	61.5	66.8	64.1	63.0
DART (EMNLP'25)	55.4	60.6	60.5	54.0	60.0	58.1	61.5	64.9	62.4	62.1
V^2Drop (CVPR'26)	58.9	61.2	62.6	56.4	65.6	61.0	63.6	67.5	52.6	62.9
HEAT (Ours)	**59.5**	**61.4**	**66.6**	**57.4**	**66.4**	**61.8**	**64.7**	**71.7**	**64.6**	**63.2**

❷Document understanding. Figure 3 shows that HEAT achieves the best performance across all eight model–benchmark combinations at 33.3% retention, including 84.6/86.4 on DocVQA and 79.6/78.6 on TextVQA for Qwen2.5-VL-7B and InternVL2.5-8B, respectively. The consistent gains on document suggest that adaptive depth is particularly beneficial when query-relevant evidence remains spatially distributed and requires additional cross-modal processing before reduction.

❸Video understanding. Table 3 shows that HEAT maintains strong performance across both backbones as retention decreases from 33.3% to 11.1%. Under the most aggressive setting, HEAT achieves 59.5/61.8 on VideoMME and 66.6/71.7 on MVBench for Qwen2.5-VL-7B and InternVL2.5-8B, respectively. The gains indicate that adapting pruning depth helps preserve temporally distributed evidence that may not be sufficiently resolved at a fixed early layer.

❹Inference efficiency. Table 2 reports efficiency at approximately 66% visual-token reduction. Compared with vanilla model, HEAT reduces LLM-prefill latency by 35.4% on MME and 36.9% on DocVQA, yielding 1.55× and 1.59× speedups, respectively. Meanwhile, throughput increases from 14.2/13.7 to 22.0/21.7 KToken/s, while prefill computation decreases from 33.5/35.0 to 19.7/19.4 TFLOPs. HEAT also reduces visual KV-cache memory by 65.9% and 65.6%, with no increase in peak GPU memory over V^2Drop. Although adaptive depth scaling introduces 2.92/2.87 ms of algorithmic overhead, it still provides substantial net inference savings.

5.2 Ablation and Sensitivity Analysis

To answer **Q2**, we separately examine the effects of pruning depth, the CRM coverage criterion, and input–depth correspondence under matched token budgets and selectors. We further study the sensitivity of the scheduler hyperparameters.

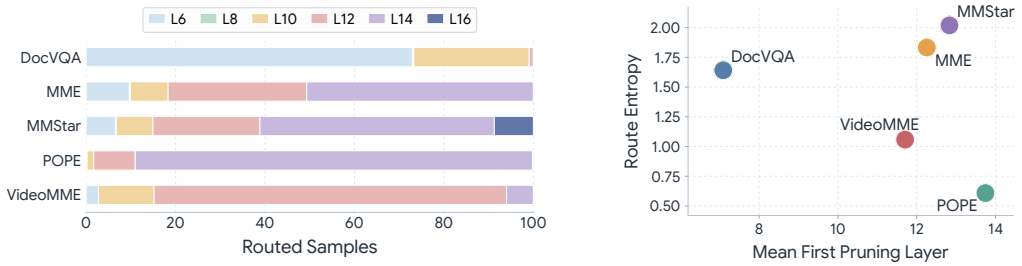


Figure 4: **Depth assignments reveal task-dependent pruning behavior and assignment diversity.** The left panel gives first pruning layer frequencies and the right reports route entropy by task.

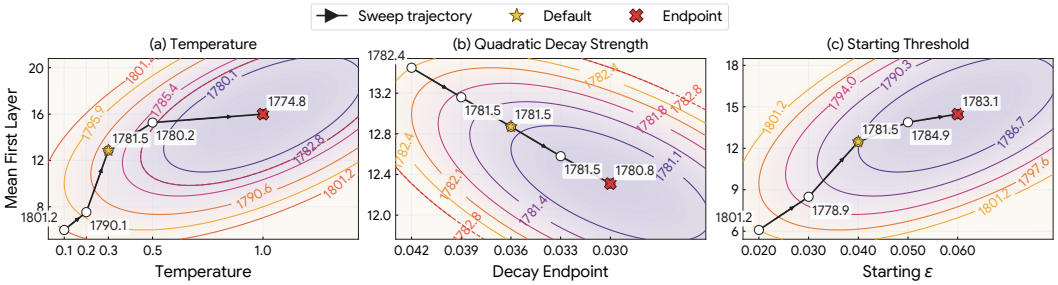


Figure 5: **Scheduler sensitivity on LLaVA-1.5-7B.** The three panels vary temperature, quadratic-decay strength, and relative-threshold scale.

❶Ablation analysis. Table 4(a) directly demonstrates the limitation of fixed-depth pruning. Different benchmarks favor different intervention layers: MMStar and POPE perform best at L16, whereas MME peaks at L14. Fixed progressive depth configurations exhibit a similar trade-off. In contrast, HEAT achieves 63.0 MMStar, 2318.4 MME, and 87.9 POPE through input-adaptive depth scaling.

Part (b) isolates the CRM criterion while keeping the depth-scaling formulation unchanged. Relative TopK mass achieves the strongest overall performance among the evaluated alternatives, supporting semantic coverage as an effective indicator of whether the current retention budget sufficiently captures query-relevant visual information.

Part (c) further tests whether assigned pruning depths must correspond to the input. Randomly shuffling depth assignments while preserving their distribution and average computation reduces performance by 0.6 MMStar, 18.8 MME, and 0.9 DocVQA, with the original assignment outperforming the shuffled counterpart in 29 of 30 trials. This confirms that the benefit comes from matching pruning depth to each input rather than from depth variation alone.

Table 4: **Ablations of pruning depth, CRM criteria, and input–depth alignment.**

(a) Pruning depth			
Configuration	MMStar	MME	POPE
Fixed L6	57.2	2285.7	85.8
Fixed L12	61.3	2302.4	87.1
Fixed L14	62.2	2315.9	87.2
Fixed L16	62.5	2298.5	87.9
Progressive $6 \rightarrow 10 \rightarrow 14$	60.9	2313.2	87.8
Progressive $12 \rightarrow 14 \rightarrow 16$	63.0	2305.3	87.5
HEAT	**63.0**	**2318.4**	**87.9**
(b) Dynamic-depth criterion			
Criterion	MMStar	MME	POPE
JSD distance	62.4	2314.2	87.6
L1 distance	62.3	2318.3	87.1
L2 distance	62.3	2311.8	87.5
Cosine distance	62.5	2311.1	87.8
Relative TopK mass (HEAT)	**63.0**	**2318.4**	**87.9**
(c) Matched-depth counterfactual			
Benchmark	Original	Shuffled	Below original
MMStar	63.0	62.4 ± 0.4	9/10
MME	2318.4	2299.6 ± 7.5	10/10
DocVQA	89.5	88.6 ± 0.2	10/10

Figure 4 further reveals task-dependent depth-scaling behavior. For example, 73.0% of DocVQA samples first prune at L6, whereas 52.4% of MMStar samples first prune at L14 and 78.8% of VideoMME samples at L12. These differences indicate that the scheduler allocates different amounts of visual processing according to the evolving relevance state of each input.

❷Hyperparameter sensitivity. Figure 5 examines how scheduler hyperparameters affect pruning depth. Varying the temperature shifts the mean first-pruning layer from 6.00 to 16.00, while the quadratic-decay parameter moves it from 13.45 to 12.31 with only a 1.6-point MME variation.

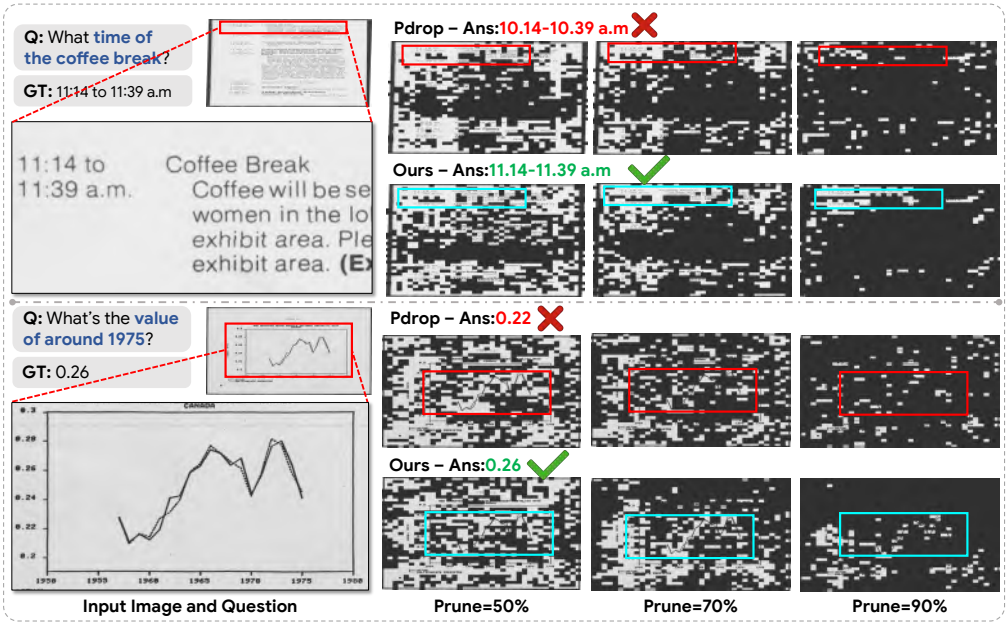


Figure 6: **Adaptive depth scaling preserves OCR text and chart trends.** Under aggressive pruning, HEAT retains the coffee-break time, chart scale, and curve trend around 1975, while PyramidDrop loses key evidence and produces incorrect answers.

Adjusting the starting threshold changes the mean first-pruning layer from 6.09 to 14.47. These results show that the scheduler provides direct control over how conservatively HEAT delays reduction while remaining stable over a broad range of settings. Additional analyses are provided in Appendix B.

5.3 Visualization

To answer **Q3**, we compare HEAT with PyramidDrop under matched pruning ratios on two document-understanding inputs that require fine-grained OCR and chart evidence.

Figure 6 shows that HEAT retains the text line needed to recover the coffee-break time even at 90% pruning, producing the correct answer of 11:14–11:39 a.m., whereas PyramidDrop predicts 10:14–10:39 a.m. In the chart example, HEAT preserves the vertical scale and the curve segment and overall trend around 1975, yielding 0.26 instead of PyramidDrop's 0.22. OCR and chart reasoning depend on small text, axis values, and structured evidence distributed across the image. By deferring a pending reduction stage when its fixed top-K budget does not yet cover sufficient relevance mass, HEAT provides additional cross-modal processing before token selection; this property is particularly useful for OCR-intensive inputs and is consistent with the document results in Figure 3. Additional retention masks are provided in Figures 9–12.

6 Conclusion

In this work, we propose HEAT, a training-free framework for adaptive visual token reduction that addresses the fixed-depth mismatch in LVLM inference. We show that visual relevance evolves across layers, making single-layer importance estimates unreliable, while pruning sensitivity varies across inputs, making a uniform intervention depth suboptimal. HEAT addresses these issues with a cross-layer relevance monitor that tracks text-aligned semantic coverage and a depth-adaptive threshold scheduler that determines input-specific pruning depth. This allocates additional visual processing to sensitive inputs while compressing inputs with lower processing demands earlier. Across Qwen2.5-VL-7B, InternVL2.5-8B, LLaVA-1.5-7B/13B, and InternVL3.5-8B-Instruct, HEAT consistently preserves performance across image, document, video, and captioning tasks. At 66.7% visual-token reduction, HEAT retains 99.4% of full-token general-image performance on average, reduces visual KV-cache memory by 66%, and achieves up to $1.59\times$ LLM-prefill speedup.

REFERENCES

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

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. In *IEEE International Conference on Computer Vision*, pp. 2425–2433, 2015.

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

Junjie Chen, Xuyang Liu, Zichen Wen, Yiyu Wang, Siteng Huang, and Honggang Chen. Variation-aware vision token dropping for faster large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3489–3499, 2026a.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *European Conference on Computer Vision*, Lecture Notes in Computer Science, pp. 19–35, 2024a.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Conference on Neural Information Processing Systems*, 2024b.

Yongru Chen, Kai Zhang, Zeliang Zong, Yuchen Lu, Wenming Tan, Ye Ren, and Jilin Hu. One layer's trash is another layer's treasure: Adaptive layer-wise visual token selection in lvlms. *arXiv preprint arXiv:2606.14277*, 2026b.

Zhe Chen, Weiyun Wu, Yusheng Tian, Zhangwei Ye, Yichi Lu, Jinguo Xiao, et al. Internvl2.5: Expanding performance boundaries with an open-source multimodal large language model. *arXiv preprint arXiv:2412.05271*, 2024c.

X.AI Corp. Grok-1.5 vision preview: Connecting the digital and physical worlds with our first multimodal model., 2024.

Mohamed Dhouib, Davide Buscaldi, Sonia Vanier, and Aymen Shabou. PACT: pruning and clustering-based token reduction for faster visual language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14582–14592, 2025.

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

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24108–24118, 2025.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pp. 148–166, 2024.

Jiawei Guo, Feifei Zhai, Pu Jian, Qianrun Wei, and Yu Zhou. Crop: Contextual region-oriented visual token pruning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9767–9783, 2025.

Xiaohu Huang, Hao Zhou, and Kai Han. Prunevid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19959–19973, 2025.

Yihong Huang, Fei Ma, Yihua Shao, Jingcai Guo, Zitong Yu, Laizhong Cui, and Qi Tian. Nüwa: Mending the spatial integrity torn by vlm token pruning. *arXiv preprint arXiv:2602.02951*, 2026.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6700–6709, 2019.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Min Joon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth A dozen images. *CoRR*, abs/1603.07396, 2016.

Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. 2017.

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

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Conference on Neural Information Processing Systems*, 2023.

Jizhihui Liu, Guangdao Zhu, and Feiyi Du. Hiprune: Training-free visual token pruning via hierarchical attention in vision-language models (student abstract). In *AAAI Conference on Artificial Intelligence*, pp. 41275–41277, 2026.

Ting Liu, Liangtao Zhu, Xiantao Wang, Wenbo Ma, Fei Zhang, and Jizhong Yin. Multi-stage vision token dropping: Towards efficient multimodal large language model. *arXiv preprint arXiv:2411.10803*, 2024a.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pp. 216–233, 2024b.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12):497–523, 2024c.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq R. Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics*, Findings of ACL, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for VQA on document images. In *IEEE Winter Conference on Applications of Computer Vision*, pp. 2199–2208, 2021.

Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Mateusz Malinowski, Yi Yang, Carl Doersch, et al. Perception test: A diagnostic benchmark for multimodal video models. In *Conference on Neural Information Processing Systems*, pp. 42748–42761, 2023.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

Yizheng Sun, Yanze Xin, Hao Li, Jingyuan Sun, Chenghua Lin, and Riza Theresa Batista-Navarro. Lvpruning: An effective yet simple language-guided vision token pruning approach for multi-modal large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4299–4308, 2025.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18992–19001, 2025.

Jintao Tong, Wenwei Jin, Pengda Qin, Anqi Li, Yixiong Zou, Yuhong Li, Yuhua Li, and Ruixuan Li. Flowcut: Rethinking redundancy via information flow for efficient vision-language models. In *Conference on Neural Information Processing Systems*, pp. 1–21, 2025.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *CoRR*, abs/2508.18265, 2025.

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. Token pruning in multimodal large language models: Are we solving the right problem? In *Findings of the Association for Computational Linguistics*, pp. 15537–15549, 2025a.

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

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

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

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *AAAI Conference on Artificial Intelligence*, pp. 22128–22136, 2025a.

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

Hao Yin, Guangzong Si, and Zilei Wang. Lifting the veil on visual information flow in mllms: Unlocking pathways to faster inference. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025*, pp. 9382–9391, 2025.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

Quan-Sheng Zeng, Yunheng Li, Qilong Wang, Peng-Tao Jiang, Zuxuan Wu, Ming-Ming Cheng, and Qibin Hou. A glimpse to compress: Dynamic visual token pruning for large vision-language models. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026.

Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Haitao Mi, and Dong Yu. Vscan: Rethinking visual token reduction for efficient large vision-language models. *Transactions on Machine Learning Research*, 2026.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. In *Conference on Neural Information Processing Systems*, pp. 1–22, 2025a.

Weichen Zhang, Zhui Zhu, Ningbo Li, Kebin Liu, and Yunhao Liu. Adaptinfer: Adaptive token pruning for vision-language model inference with dynamical text guidance. *arXiv preprint arXiv:2508.06084*, 2025b.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*, pp. 1–18, 2025c.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: benchmarking multi-task long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13691–13701, 2025.

Xin Zou, Di Lu, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Xu Zheng, Linfeng Zhang, and Xuming Hu. Don't just chase "highlighted tokens" in mllms: Revisiting visual holistic context retention. In *Conference on Neural Information Processing Systems*, pp. 1–34, 2025.

Appendix Contents

Appendix A: Additional Method and Complexity Details 14
- Appendix A.1: Execution Details 14
- Appendix A.2: Computational Complexity 14

Appendix B: Additional Experiments 15
- Appendix B.1: Experimental Setup Details 15
- Appendix B.2: Additional Token-Budget Results 18
- Appendix B.3: Full Cross-Model OCR Results 18
- Appendix B.4: LLaVA Family Results 18
  - Appendix B.4.1: LLaVA-1.5-7B General-Image Evaluation 18
  - Appendix B.4.2: Cross-Scale LLaVA-1.5-13B Evaluation 18
- Appendix B.5: Transfer to InternVL3.5-8B-Instruct 21
- Appendix B.6: Open-Ended Caption Generation 21
- Appendix B.7: Scheduler Sensitivity 23
- Appendix B.8: Additional Qualitative Visualizations 24

AI Use Statement 24

Ethics Statement 25

Reproducibility Statement 25

All appendix entries and page numbers are hyperlinked.

A Additional Method and Complexity Details

A.1 Execution Details

After each selection, HEAT restores the retained tokens to their original order and uses the same indices to compact the hidden sequence, attention mask, positional metadata, visual spans, and materialized KV cache. Removed positions therefore participate in neither subsequent attention nor cache storage. The backbone retains FlashAttention for its standard forward pass.

A.2 Computational Complexity

We account for both the sequence-length changes induced by pruning and the overhead of depth-adaptive monitoring and token selection. Let x denote the sequence length, d the hidden width, and d_{ff} the feed-forward width. The FLOPs of one standard Transformer block are

$$\mathcal{C}_{blk}(x) = 4xd^2 + 2x^2d + 2xd\,d_{ff}. \tag{11}$$

For an L-layer model, let b be the block after which a single reduction changes the total sequence length from n to $\widetilde{n}$, and let $\mathcal{C}_{red}$ denote its reduction overhead. The resulting prefill cost is

$$\mathcal{C}_{single} = b\,\mathcal{C}_{blk}(n) + \mathcal{C}_{red} + (L-b)\mathcal{C}_{blk}(\widetilde{n}). \tag{12}$$

Equation 12 charges the blocks before and after reduction at their respective sequence lengths and explicitly includes the reduction overhead.

HEAT extends this accounting to input-adaptive, multi-stage execution. For an input $\mathbf{x}$, let $J(\mathbf{x})$ be the number of actual pruning operations and let $0 = \ell_0 < \ell_1 < \cdots < \ell_{J(\mathbf{x})} < \ell_{J(\mathbf{x})+1} = L$ denote their layer boundaries. Let N_t be the text-token count and $N_v^{(j)}(\mathbf{x})$ the visual-token count after the jth operation, with $N_v^{(0)}(\mathbf{x})$ denoting the initial visual-token count. If $\mathcal{C}_{over}(\mathbf{x})$ denotes the cost of monitoring, selection, and sequence compaction, the prefill cost is

$$\mathcal{C}_{prefill}(\mathbf{x}) = \sum_{j=0}^{J(\mathbf{x})} (\ell_{j+1} - \ell_j)\,\mathcal{C}_{blk}\Big(N_t + N_v^{(j)}(\mathbf{x})\Big) + \mathcal{C}_{over}(\mathbf{x}). \tag{13}$$

Each block is charged according to the sequence length it actually processes. Earlier pruning reduces computation in more subsequent blocks, whereas deferring a stage retains the current visual tokens for additional LLM blocks. Although every input has the same M conceptual stages, the number of

actual pruning operations $J(\mathbf{x})$ can be smaller because the fallback operation merges all unfinished stages.

The additional cost comprises CRM evaluation at monitored layers and token selection and compaction at activated layers. Let $\mathcal{L}_{mon}(\mathbf{x})$ and $\mathcal{L}_{act}(\mathbf{x})$ denote these two sets, respectively, and let K_l denote the target token count used at layer l for the pending stage or fallback operation. We define the per-layer monitoring and activation costs as $\mathcal{C}_l^{mon} = \mathcal{O}(N_l N_t d + N_l \log K_l)$ and $\mathcal{C}_l^{act} = \mathcal{O}(N_l N_t d + N_l \log K_l + K_l d)$. A conservative upper bound is

$$\mathcal{C}_{over}(\mathbf{x}) = \sum_{l \in \mathcal{L}_{mon}(\mathbf{x})} \mathcal{C}_l^{mon} + \sum_{l \in \mathcal{L}_{act}(\mathbf{x})} \mathcal{C}_l^{act}. \tag{14}$$

The first sum covers text-aligned relevance, normalization, and the CRM top-K mass computation. The second covers the selector's text-to-image scoring, top-K operation, and sequence/KV compaction when a stage is activated; the scalar DTS comparison is absorbed into these terms. Our configuration evaluates the stage trigger at up to five monitored layers and uses at most three actual pruning operations.

Table 2 includes this overhead. Monitoring, selection, and compaction take 2.92 and 2.87 ms for image and OCR inputs, respectively, while the complete LLM prefill achieves $1.55\times$ and $1.59\times$ speedups over the full-token model. These measurements include the additional work introduced by adaptive depth scaling.

B Additional Experiments

We provide the full experimental configuration, aggressive token-budget results, cross-model OCR comparisons, evaluations on LLaVA and InternVL3.5, open-ended captioning results, scheduler sensitivity, and additional qualitative examples.

B.1 Experimental Setup Details

We first describe the benchmarks, models, baselines, evaluation protocol, and depth-scaling configurations used throughout the experiments.

B.1.1 Evaluation Benchmarks

General-image understanding. We evaluate broad visual perception and reasoning on ten benchmark variants:

- **MME** (Fu et al., 2023) evaluates both perception and cognition through 14 subtasks and uses concise instruction–answer pairs to reduce ambiguity in automatic evaluation.
- **MMBench-EN/CN** (Liu et al., 2024b) organizes multimodal ability into a coarse-to-fine hierarchy covering perception, reasoning, and fine-grained subskills. MMBench-CN provides the corresponding Chinese-language evaluation.
- **POPE** (Li et al., 2023) evaluates object hallucination through binary questions about whether an object is present, reporting accuracy, precision, recall, and F1 to quantify visual faithfulness.
- **GQA** (Hudson & Manning, 2019) combines real images, scene graphs, and compositional questions with dense object and relation annotations, testing fine-grained visual reasoning and scene understanding.
- **VQAv2** (Antol et al., 2015) evaluates open-ended visual question answering on real-world images. Its balanced question pairs reduce answer-only language priors and emphasize grounding in visual evidence.
- **MMStar** (Chen et al., 2024b) contains 1,500 human-verified, vision-essential samples spanning six core capabilities and 18 fine-grained dimensions. Samples answerable without visual input are explicitly filtered.
- **RealWorldQA** (Corp, 2024) evaluates physical and spatial reasoning in realistic environments, testing whether a model grounds its answers in everyday visual scenes.

- **BLINK** (Fu et al., 2024) targets core visual-perception skills through visually grounded questions that require fine-grained comparison and spatial understanding.
- **AI2D** (Kembhavi et al., 2016) evaluates understanding and reasoning over scientific diagrams, including the relationships among textual labels, visual elements, and diagram structure.

Chart and document understanding. We evaluate dense textual, numerical, and layout reasoning on four benchmarks:

- **TextVQA** (Singh et al., 2019) evaluates reasoning over scene text embedded in Open Images (Krasin et al., 2017), such as signs, labels, and product packaging. We use its validation split.
- **DocVQA** (Mathew et al., 2021) evaluates document understanding with 50,000 questions over 12,767 document images and includes multiple reasoning categories for fine-grained analysis.
- **OCRBench** (Liu et al., 2024c) aggregates 29 datasets spanning text recognition, scene-text VQA, key-information extraction, and handwritten mathematical expression recognition.
- **ChartQA** (Masry et al., 2022) evaluates visual and logical reasoning over charts using both human-written and machine-generated questions. We report results on the standard test split.

Image captioning. We evaluate open-ended image description on:

- **Flickr30K** (Young et al., 2014) contains 31,783 images, each paired with five English captions. We use it to evaluate caption quality under visual-token reduction.

Video understanding. We evaluate temporal and long-context reasoning on four benchmark families; VideoMME is reported both with and without subtitles:

- **VideoMME** (Fu et al., 2025) contains 900 videos from 30 domains with durations ranging from short clips to hour-long content and provides 2,700 expert-annotated question–answer pairs. Reporting both subtitle settings separates visual-temporal reasoning from assistance provided by transcripts.
- **MVBench** (Li et al., 2024) evaluates multiple video-understanding skills, including action, event, and temporal reasoning, under a unified multiple-choice protocol.
- **MLVU** (Zhou et al., 2025) targets long-video understanding over videos lasting up to two hours and covers both holistic comprehension and fine-grained detail reasoning.
- **PerceptionTest** (Patraucean et al., 2023) evaluates visual perception and temporal reasoning over real-world videos, complementing the long-form focus of VideoMME and MLVU.

B.1.2 Models

We conduct the main evaluation with Qwen2.5-VL-7B-Instruct (Bai et al., 2025) and InternVL2.5-8B (Chen et al., 2024c). The LLaVA family, LLaVA-1.5-7B and LLaVA-1.5-13B (Liu et al., 2023), provides projector-based and cross-scale transfer evaluations. We further evaluate InternVL3.5-8B-Instruct (Wang et al., 2025) to test transfer to a newer backbone. These models cover distinct visual encoders, multimodal connectors, language backbones, and visual sequence lengths, allowing us to evaluate adaptive depth scaling across different LVLM architectures.

B.1.3 Baselines

Vanilla. The unpruned backbone retains every visual token and provides the full-token reference for both accuracy and efficiency.

Random. Random pruning retains a uniformly sampled subset under the same terminal token budget, providing a selector-independent lower control.

FastV (Chen et al., 2024a) uses language-model attention to identify and remove redundant visual tokens at an early transformer layer.

Table 5: **Depth-scaling configurations across models and token-reduction budgets.** Rows specify the stage-wise keep/drop factors and the relative-margin endpoints at the first and last monitored layers.

Model	Reduction r	Stage drop s_r	Stage retention γ_r	δ_r: L6 $\rightarrow$ L14
Qwen2.5-VL / InternVL / LLaVA-1.5-13B	0.667	0.30687	0.69313	$0.070 \rightarrow 0.035$
	0.778	0.39450	0.60550	$0.090 \rightarrow 0.045$
	0.889	0.51941	0.48059	$0.120 \rightarrow 0.060$
LLaVA-1.5-7B	0.667	0.30687	0.69313	$0.042 \rightarrow 0.036$
	0.778	0.39450	0.60550	$0.054 \rightarrow 0.046$
	0.889	0.51941	0.48059	$0.072 \rightarrow 0.062$

Table 6: **Layer-wise relative margins and corresponding coverage thresholds for Qwen2.5-VL, InternVL, and LLaVA-1.5-13B.** The upper and lower blocks report relative margins and the resulting absolute thresholds, respectively.

Quantity	Reduction r	L6	L8	L10	L12	L14
$\delta_r(l)$	0.667	0.07000	0.06781	0.06125	0.05031	0.03500
	0.778	0.09000	0.08719	0.07875	0.06469	0.04500
	0.889	0.12000	0.11625	0.10500	0.08625	0.06000
$\tau_r(l) = \gamma_r[1 + \delta_r(l)]$	0.667	0.74165	0.74013	0.73558	0.72800	0.71739
	0.778	0.66000	0.65830	0.65319	0.64467	0.63275
	0.889	0.53826	0.53646	0.53105	0.52204	0.50942

PyramidDrop (Xing et al., 2024) distributes reduction across multiple transformer layers to form a pyramidal visual-token schedule.

DivPrune (Alvar et al., 2025) formulates token selection as a max–min diversity problem and retains a subset with high pairwise diversity.

DART (Wen et al., 2025b) removes redundancy by selecting visual tokens that are dissimilar to a small set of pivot tokens rather than relying only on importance ranking.

V^{2}Drop (Chen et al., 2026a) uses variation-aware signals to progressively remove redundant visual tokens across transformer layers.

Evaluation protocol. Unless otherwise stated, all methods use the same backbone, native input resolution, terminal token budget, and deterministic decoding. The main general-image results use 33.3% and 22.2% average retention, and the appendix adds the 11.1% setting for Qwen and LLaVA. Video results cover all three budgets. The main document comparison uses 33.3% retention, while the supplementary results extend it to 22.2% and 11.1%. MME is reported as a raw perception-plus-cognition score; all other accuracy metrics are percentages, and – denotes an unavailable matched-protocol result. Mechanism controls use HEAT's text-to-image attention selector.

Efficiency protocol. Efficiency is averaged over three runs on 100 fixed-resolution samples per modality. LLM-prefill time excludes preprocessing, vision encoding, and autoregressive decoding.

Depth-scaling configuration. All models and budgets use the layer set $\mathcal{L} = \{4, 6, 8, 10, 12, 14, 16\}$, $M = 3$ conceptual reduction stages, the CRM relevance measure defined in Section 4.1, and temperature $T = 0.3$. No pruning occurs at layer 4. Layers 6–14 are the five monitored layers at which Equation 8 is evaluated, and layer 16 is the fallback layer that completes all unfinished stages without an additional coverage test. The budget $K^{(m+1)}$ of a pending stage remains fixed while these monitored layers are scanned. At the fallback layer, any remaining stages are combined according to Equation 10.

For each reduction ratio, Table 5 reports the relative-margin endpoints at layers 6 and 14; the intermediate margins follow the quadratic interpolation in Equation 7. Except for LLaVA-1.5-7B, all model families use the same budget-specific endpoint pairs. For reproduction, $\gamma_r = (1-r)^{1/M}$ is the per-stage retention factor, $s_r = 1 - \gamma_r$ is the corresponding drop fraction, and $\tau_r(l) = \gamma_r[1 + \delta_r(l)]$ is the coverage threshold. Table 6 lists the resulting layer-wise values.

LLaVA-1.5-7B reproduction details. The LLaVA-1.5-7B image results use native 336×336 resolution, batch size one, and the model-specific schedules in Table 5. Benchmark scores are taken

directly from the official distributed `lmms-eval` result aggregation rather than recomputed from sample logs. The pruning-depth statistics aggregate 254,296 model calls: 254,218 evaluation examples and 78 replicas introduced by eight-rank sampler padding. The default selector uses text-to-image attention, and monitoring and pruning are disabled during autoregressive decoding.

B.2 Additional Token-Budget Results

The main general-image table reports Qwen and InternVL at 33.3% and 22.2% average token retention, while the main video table covers all three budgets. Here we report the omitted aggressive Qwen general-image budget using the same models, selectors, and evaluation protocol.

B.2.1 Aggressive General-Image Budget

Table 7: **Qwen2.5-VL-7B general-image accuracy at the aggressive 11.1% visual-token budget.** All reducers use the same retained-token budget.

Method	MME	MMB(EN)	MMB(CN)	MMStar	POPE	RWQA	BLINK	AI2D	GQA	VQAv2
Average Retain 11.1% Tokens										
Random	2065.0	73.8	69.3	46.4	81.2	59.3	43.2	70.8	54.5	72.0
FastV (ECCV'24)	1961.4	73.7	67.6	43.8	70.7	51.5	49.0	69.9	48.6	66.6
PyramidDrop (CVPR'25)	2176.2	76.8	73.3	49.1	85.1	65.8	51.7	74.3	53.8	76.9
DivPrune (CVPR'25)	2112.2	76.8	74.2	49.0	83.7	62.2	43.5	73.3	56.5	75.1
DART (EMNLP'25)	1996.8	76.9	73.0	45.0	79.4	55.7	44.2	72.5	52.9	70.7
V^2Drop (CVPR'26)	2245.2	77.2	76.9	53.3	81.8	62.1	52.1	73.2	54.8	73.4
HEAT (Ours)	**2259.7**	**78.6**	**77.1**	**55.3**	**87.0**	**67.8**	**52.3**	**74.9**	**57.5**	**79.1**

Table 7 shows that HEAT has the highest score on all ten benchmarks at 11.1% retention. It obtains 2259.7 MME and 55.3 MMStar, compared with the best competing scores of 2245.2 and 53.3, respectively. Under the same terminal token budget, this result extends the main general-image comparison to the most aggressive evaluated setting.

B.3 Full Cross-Model OCR Results

Figure 7 shows the complete document comparison across all baselines, backbones, and budgets. At 22.2% retention, HEAT has the highest score in seven of eight model–benchmark pairs; the exception is InternVL2.5-8B on OCRBench, where DART obtains 67.0 and HEAT obtains 63.7. This comparison extends the main document results to a tighter matched token budget while making the single exception explicit.

B.4 LLaVA Family Results

The main paper prioritizes matched Qwen and InternVL results at two operating budgets. Here we report the complete LLaVA-1.5-7B evaluation, test cross-scale transfer on LLaVA-1.5-13B using the common schedule in Table 5, and provide document results for both model sizes.

B.4.1 LLaVA-1.5-7B General-Image Evaluation

Table 8 shows that HEAT retains much of the full-token performance on LLaVA-1.5-7B as retention decreases to 11.1%. It obtains 1780.8 MME, compared with 1782.8 for the unpruned model and 1747.5 for the best competing reducer, while GQA and VQAv2 remain within 0.2 points of their full-token scores. These matched-budget results extend the evaluation of adaptive depth scaling to a projector-based backbone. The complete configuration is given in Appendix B.1.

B.4.2 Cross-Scale General-Image Evaluation

Table 8 reports the corresponding results on the 13B backbone. At 11.1% retention, HEAT obtains 1658.5 MME, compared with 1665.0 for the unpruned model and 1636.5 for the best competing reducer. The monitored layers, CRM, and selector are unchanged, and HEAT uses the family-level

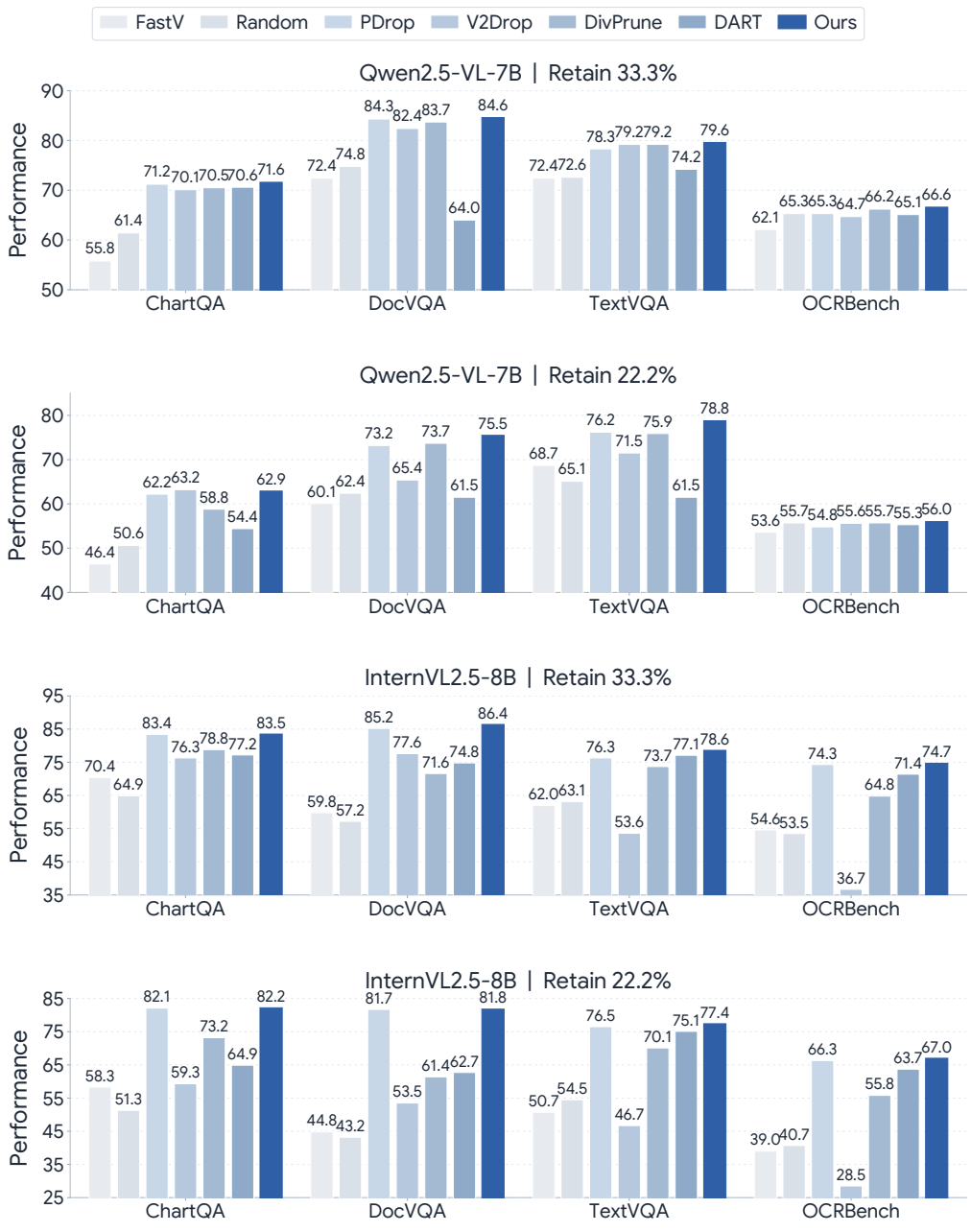


Figure 7: **Full document-understanding comparison across backbones, baselines, and visual-token budgets.** This figure extends the main-paper view with every evaluated retention setting.

threshold schedule in Table 5. These results show that the same depth-scaling formulation can be applied at the 13B scale without changing the method.

B.4.3 Document Understanding

Table 9 shows the document results across both LLaVA scales. At 22.2% retention, HEAT reaches 23.4/27.9 DocVQA on the 7B/13B models, compared with their full-token scores of 23.8/28.2; at 11.1%, it obtains 22.5/26.8. HEAT has the highest reported score in all eight model–benchmark comparisons at both 22.2% and 11.1% retention, extending the matched-budget document results across model scales.

Table 8: **LLaVA-1.5 general-image accuracy across 7B and 13B model scales and visual-token budgets.** Row blocks separate backbone scale and retention level.

Method	MME	MMB(EN)	MMB(CN)	MMStar	POPE	RWQA	BLINK	AI2D	GQA	VQAv2
LLaVA-1.5-7B, *Upper Bound, All Tokens (100%)*										
Upper Bound	1782.8	62.1	50.3	32.7	85.9	55.6	38.5	52.5	60.6	75.8
Average Retain 33.3% Tokens										
Random	1702.1	58.2	46.3	31.6	83.4	52.3	38.3	52.6	59.6	73.4
FastV (ECCV'24)	1635.3	59.6	45.7	31.7	80.4	53.1	38.3	52.5	56.6	70.0
PyramidDrop (CVPR'25)	1764.0	61.7	50.0	32.4	84.6	55.1	38.5	52.3	59.8	75.4
DivPrune (CVPR'25)	1734.4	60.0	48.8	31.7	85.6	53.7	38.1	52.7	60.2	75.1
DART (EMNLP'25)	1777.0	60.9	49.1	31.6	82.9	55.2	38.3	**53.6**	58.9	74.3
V^2Drop (CVPR'26)	1777.4	60.8	49.7	31.6	84.7	54.6	38.6	53.4	59.8	74.1
HEAT (Ours)	**1781.5**	**61.9**	**50.1**	**32.9**	**86.0**	**55.7**	**38.6**	52.6	**60.6**	**75.8**
Average Retain 22.2% Tokens										
Random	1648.1	57.6	44.5	32.1	82.4	50.1	37.9	52.2	59.1	71.7
FastV (ECCV'24)	1559.0	57.0	42.8	29.9	76.7	49.8	37.7	52.1	54.7	66.7
PyramidDrop (CVPR'25)	1769.9	61.5	49.7	32.6	84.0	55.0	38.4	51.7	59.2	74.8
DivPrune (CVPR'25)	1679.8	58.7	46.7	32.2	85.8	53.3	38.4	52.3	59.7	74.4
DART (EMNLP'25)	1724.2	60.4	48.6	31.2	81.1	54.0	38.1	52.7	58.1	73.1
V^2Drop (CVPR'26)	1764.9	60.1	49.4	31.9	84.4	53.7	38.4	52.4	59.1	73.1
HEAT (Ours)	**1779.8**	**62.0**	**50.3**	**33.0**	**86.0**	**55.6**	**38.5**	**52.6**	**60.6**	**75.8**
Average Retain 11.1% Tokens										
Random	1512.9	53.1	38.3	31.8	78.6	48.5	38.0	51.7	56.2	67.9
FastV (ECCV'24)	1310.3	48.7	34.7	27.2	66.3	45.8	37.9	51.4	48.3	57.5
PyramidDrop (CVPR'25)	1723.3	60.0	48.5	31.2	82.3	54.2	38.3	52.0	57.4	73.6
DivPrune (CVPR'25)	1646.3	56.9	44.4	31.0	84.4	51.4	38.0	52.0	58.5	72.5
DART (EMNLP'25)	1647.0	58.2	45.6	30.9	77.1	51.8	38.1	52.4	55.1	70.2
V^2Drop (CVPR'26)	1747.5	59.5	48.6	31.7	82.5	52.3	38.4	52.7	58.3	71.0
HEAT (Ours)	**1780.8**	**61.9**	**50.2**	**32.9**	**85.8**	**55.9**	**38.5**	**52.7**	**60.5**	**75.6**
LLaVA-1.5-13B, *Upper Bound, All Tokens (100%)*										
Upper Bound	1665.0	66.0	54.8	35.3	85.5	56.3	42.0	56.2	61.9	77.8
Average Retain 33.3% Tokens										
Random	1623.7	63.1	51.9	34.2	83.6	51.1	41.3	54.8	60.2	75.1
FastV (ECCV'24)	1531.6	64.2	53.5	35.4	80.7	55.0	41.9	55.2	57.1	71.0
PyramidDrop (CVPR'25)	1664.2	66.2	54.6	35.0	85.4	56.1	41.4	55.7	61.4	77.3
DivPrune (CVPR'25)	1622.6	64.3	52.7	34.5	85.4	55.3	41.1	56.0	61.1	76.6
DART (EMNLP'25)	1653.4	66.0	53.8	35.4	84.3	54.6	41.7	55.9	60.4	76.8
V^2Drop (CVPR'26)	1673.0	65.8	54.3	34.7	83.8	55.4	41.7	55.9	60.6	75.6
HEAT (Ours)	**1679.4**	**66.2**	**54.7**	**35.6**	**85.6**	**56.3**	**41.9**	**56.3**	**61.8**	**77.7**
Average Retain 22.2% Tokens										
Random	1591.0	61.5	48.9	33.7	83.0	50.1	41.8	55.1	59.0	73.3
FastV (ECCV'24)	1474.6	62.7	52.5	34.3	77.2	52.8	41.6	55.5	54.9	67.8
PyramidDrop (CVPR'25)	1654.3	66.1	54.3	35.1	85.4	55.2	41.1	55.6	60.7	77.0
DivPrune (CVPR'25)	1598.3	63.3	51.7	34.2	84.2	55.6	41.3	56.1	60.2	75.9
DART (EMNLP'25)	1648.2	64.9	53.4	34.6	83.4	54.0	41.7	55.7	59.1	75.7
V^2Drop (CVPR'26)	1662.1	65.6	54.0	34.5	83.1	54.8	41.8	55.9	59.7	74.1
HEAT (Ours)	**1664.2**	**66.1**	**54.8**	**35.5**	**85.6**	**56.6**	**41.8**	**56.3**	**61.7**	**77.7**
Average Retain 11.1% Tokens										
Random	1540.9	56.8	44.8	32.0	80.0	49.5	41.3	54.9	56.3	69.4
FastV (ECCV'24)	1293.1	57.2	47.5	32.7	70.0	47.2	40.4	54.0	50.1	59.6
PyramidDrop (CVPR'25)	1628.5	65.1	53.6	34.6	84.2	54.9	41.5	55.1	59.4	75.8
DivPrune (CVPR'25)	1535.3	61.5	49.9	32.7	85.4	53.1	41.8	55.3	58.8	74.0
DART (EMNLP'25)	1572.1	62.4	51.0	32.8	80.0	54.5	40.7	54.8	56.3	72.4
V^2Drop (CVPR'26)	1636.5	65.0	52.9	34.1	81.4	52.9	41.4	55.5	57.9	71.7
HEAT (Ours)	**1658.5**	**66.2**	**54.8**	**35.6**	**85.8**	**56.5**	**41.9**	**56.2**	**61.2**	**77.4**

Table 9: **LLaVA-1.5 document-understanding accuracy across model scales and visual-token budgets.** Row blocks separate backbone scale and retention level.

Method	LLaVA-1.5-7B				LLaVA-1.5-13B			
	ChartQA	DocVQA	TextVQA	OCRBench	ChartQA	DocVQA	TextVQA	OCRBench
	Upper Bound, All Tokens (100%)				*Upper Bound, All Tokens (100%)*			
Upper Bound	17.8	23.8	48.0	20.1	19.0	28.2	52.9	23.0
	Average Retain 33.3% Tokens				*Average Retain 33.3% Tokens*			
Random	14.8	15.4	35.7	15.1	13.2	17.1	38.5	16.7
FastV (ECCV'24)	16.5	18.0	39.2	15.7	14.8	21.3	40.4	15.7
PyramidDrop (CVPR'25)	17.7	22.2	47.5	20.0	17.5	27.6	52.6	**22.6**
DivPrune (CVPR'25)	17.0	20.9	45.2	18.4	16.0	24.1	49.6	20.5
DART (EMNLP'25)	17.1	21.5	45.5	19.1	17.2	25.0	51.6	22.2
V^2Drop (CVPR'26)	14.8	15.9	39.0	14.6	12.0	18.1	45.0	17.8
HEAT (Ours)	**18.2**	**23.6**	**48.2**	**20.1**	**18.3**	**28.2**	**52.8**	22.5
	Average Retain 22.2% Tokens				*Average Retain 22.2% Tokens*			
Random	14.5	12.9	30.7	12.9	11.7	14.2	33.5	14.0
FastV (ECCV'24)	15.8	16.0	35.3	14.3	12.9	18.3	35.8	14.1
PyramidDrop (CVPR'25)	17.3	20.7	46.9	18.8	16.6	25.4	51.9	21.2
DivPrune (CVPR'25)	16.6	18.9	43.8	17.6	14.4	21.5	47.1	19.8
DART (EMNLP'25)	17.0	19.8	44.0	18.2	16.4	21.6	49.5	20.6
V^2Drop (CVPR'26)	14.5	14.0	35.8	12.7	10.8	15.5	40.6	15.3
HEAT (Ours)	**18.1**	**23.4**	**47.9**	**19.0**	**17.4**	**27.9**	**52.7**	**22.4**
	Average Retain 11.1% Tokens				*Average Retain 11.1% Tokens*			
Random	13.7	10.9	23.2	10.0	9.4	11.9	25.5	10.7
FastV (ECCV'24)	14.5	12.8	25.2	9.1	10.2	14.2	25.6	8.6
PyramidDrop (CVPR'25)	17.2	17.7	45.5	16.6	14.7	21.3	49.9	18.2
DivPrune (CVPR'25)	15.8	15.5	40.9	15.8	12.7	17.7	43.2	17.6
DART (EMNLP'25)	16.4	15.9	40.7	16.7	13.5	17.0	44.8	17.3
V^2Drop (CVPR'26)	13.7	11.7	30.5	10.7	9.6	12.1	32.0	12.6
HEAT (Ours)	**17.6**	**22.5**	**47.5**	**17.1**	**15.6**	**26.8**	**52.0**	**20.2**

B.5 Transfer to InternVL3.5-8B-Instruct

We further evaluate HEAT on InternVL3.5-8B-Instruct without model-specific threshold calibration. HEAT uses the common depth-scaling and selection configuration in Appendix B.1.

Table 10: **Transfer of HEAT to InternVL3.5-8B-Instruct across general-image and document tasks.** HEAT and FastV are evaluated under the same backbone and token budget.

Method	General-Image Understanding							Document Understanding		
	MME	MMB-E	MMB-C	MMStar	POPE	SQA	RWQA	ChartQA	DocVQA	TextVQA
	Upper Bound, All Tokens (100%)									
Upper Bound	2341.1	83.8	80.2	64.3	89.7	97.5	62.0	86.6	91.0	76.4
	Average Retain 33.3% Tokens									
FastV (ECCV'24)	2347.6	83.2	79.3	61.8	89.3	**97.3**	**61.8**	81.4	79.0	**75.5**
HEAT (Ours)	**2362.7**	**83.6**	**80.0**	**63.8**	**89.6**	97.2	61.7	**84.6**	**86.8**	75.4
	Average Retain 22.2% Tokens									
FastV (ECCV'24)	2334.0	82.8	78.4	59.8	89.2	96.7	61.8	73.2	65.0	74.1
HEAT (Ours)	**2340.8**	**83.0**	**79.3**	**63.0**	**89.4**	**97.0**	**62.2**	**81.6**	**81.3**	**74.4**
	Average Retain 11.1% Tokens									
FastV (ECCV'24)	2307.3	81.0	76.4	58.4	87.9	93.6	60.5	47.0	44.4	70.1
HEAT (Ours)	**2337.1**	**82.6**	**78.3**	**60.4**	**89.4**	**96.5**	**62.7**	**69.8**	**66.3**	**71.5**

Table 10 compares HEAT and FastV on InternVL3.5-8B-Instruct at three budgets. At $r = 0.889$, HEAT exceeds FastV by 2.0 MMStar, 22.8 ChartQA, and 21.9 DocVQA points; at $r = 0.667$, its MMStar score is within 0.5 points of the unpruned model while retaining one third of the visual tokens. Because HEAT uses the common configuration without backbone-specific threshold calibration, these results support its application to a newer InternVL backbone.

B.6 Open-Ended Caption Generation

We evaluate open-ended caption generation beyond the question-answering tasks used in the main evaluation. Figure 8 compares four backbones at three retained-token budgets.

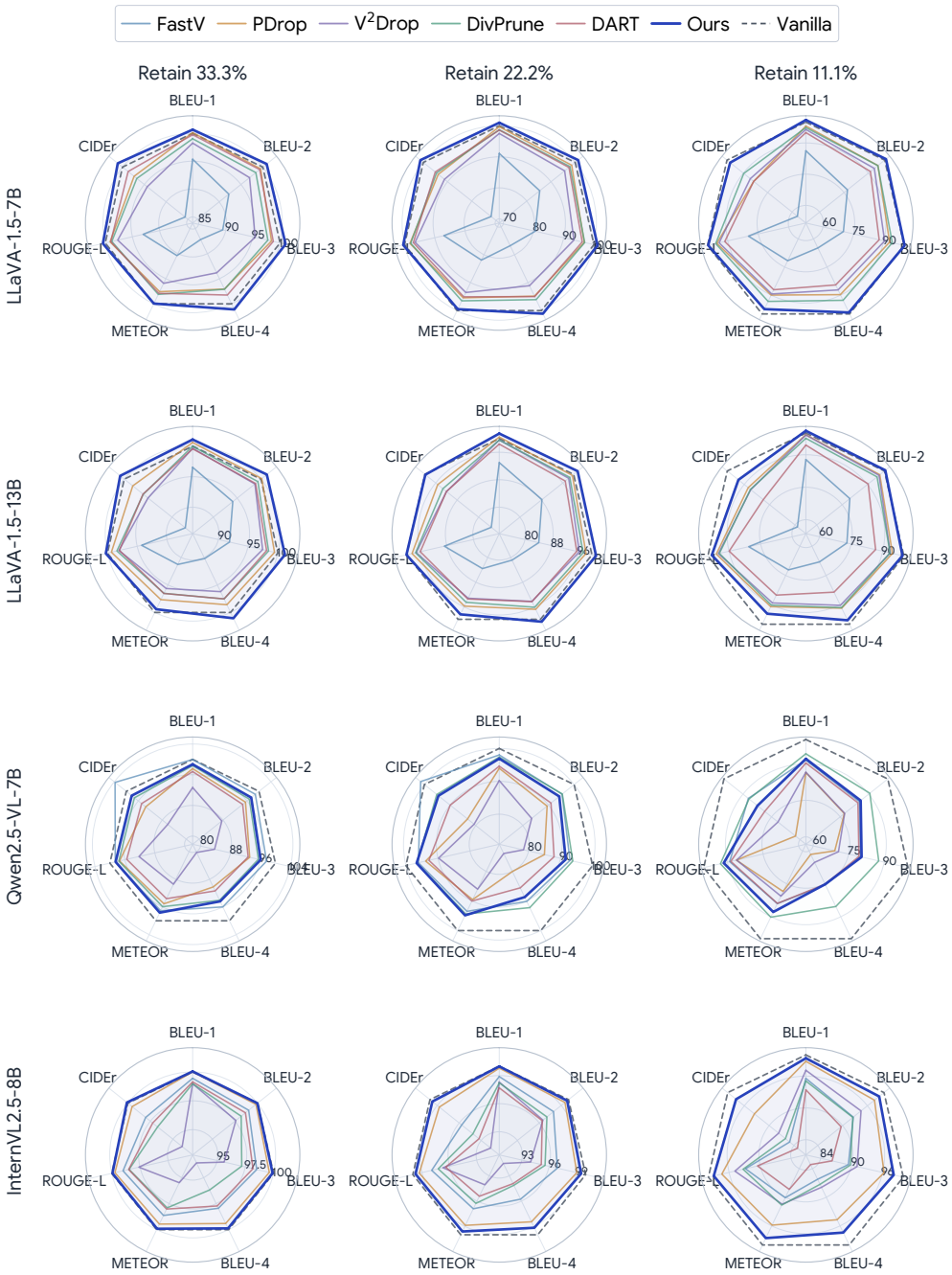


Figure 8: **Flickr30K caption-generation performance across backbones and visual-token budgets.** Panels separate model families and retained-token budgets.

Figure 8 reports caption-generation performance across backbones and retention budgets. At 33.3% and 22.2% retention, HEAT is generally among the reducers closest to the full-token result across the seven captioning metrics. At 11.1%, the gap to the full-token model becomes larger, particularly for Qwen2.5-VL and InternVL2.5, showing the effect of the terminal token budget rather than uniformly attributing the result to pruning depth.

B.7 Scheduler Sensitivity

B.7.1 Schedule-Shape Sensitivity

Table 11 isolates schedule shape by holding both mass endpoints fixed. Quadratic decay obtains 63.0 MMStar, compared with 62.9 for linear and 62.6 for cubic decay. The quadratic schedule keeps a larger margin at intermediate shallow layers than linear interpolation and relaxes more rapidly near the fallback layer, matching the intended asymmetric treatment of early and late pruning. The differences are small, with quadratic exceeding linear by 0.1 point and cubic by 0.4 point.

Table 11: **MMStar sensitivity to the threshold-schedule shape under matched top-K mass endpoints.** Linear, cubic, and quadratic schedules share the same $.80 \rightarrow .74$ endpoints.

Decay shape	Mass endpoints	MMStar
Linear	$.80 \rightarrow .74$	62.9
Cubic	$.80 \rightarrow .74$	62.6
Quadratic (Default)	$.80 \rightarrow .74$	**63.0**

B.7.2 Hyperparameter Sensitivity on LLaVA-1.5-7B

Figure 5 visualizes the relationship between MME and the mean first-pruning layer, while Table 12 provides the exact scores and fallback-completion statistics. We evaluate temperature, quadratic-decay strength, and relative-threshold scale at $r = 0.667$ on MMStar, MME, and POPE. MME is a raw score, whereas MMStar and POPE are percentages. All runs use native 336×336 resolution and batch size one, and the depth statistics cover 12,874 examples. The sweep is centered on the LLaVA-1.5-7B default configuration, $T = 0.3$ and $0.042 \rightarrow 0.036$, reported in Table 5. L16 completion denotes that the fallback operation finishes at least one remaining stage.

Table 12: **LLaVA-1.5-7B accuracy and pruning-depth behavior across scheduler hyperparameters.** Parts (a)–(c) sweep temperature, decay strength, and relative-threshold scale.

Setting	Value / schedule	MMStar	MME	POPE	Mean first-pruning layer	L16 completion (%)
			(a) Temperature			
Temperature	0.1	33.0	1801.2	85.8	6.00	0.0
	0.2	33.1	1790.1	85.8	7.54	4.2
	0.3 **(Default)**	32.9	1781.5	86.0	12.87	98.2
	0.5	32.9	1780.2	86.0	15.27	100.0
	1.0	33.0	1774.8	86.0	16.00	100.0
			(b) Quadratic decay strength			
No decay	$0.042 \rightarrow 0.042$	33.2	1782.4	86.0	13.45	99.4
Weak decay	$0.042 \rightarrow 0.039$	33.2	1781.5	86.0	13.16	99.0
Default	$0.042 \rightarrow 0.036$	32.9	1781.5	86.0	12.87	98.2
Strong decay	$0.042 \rightarrow 0.033$	32.9	1781.5	86.0	12.58	97.3
Very strong decay	$0.042 \rightarrow 0.030$	33.0	1780.8	86.0	12.31	96.1
			(c) Relative-threshold scale			
Start δ_r	$0.020 \rightarrow 0.0171$	33.0	1801.2	85.8	6.09	0.0
	$0.030 \rightarrow 0.0257$	33.2	1778.9	85.8	8.49	13.7
	$0.040 \rightarrow 0.0343$	33.0	1781.5	86.0	12.47	95.3
	$0.050 \rightarrow 0.0429$	33.2	1784.9	86.0	13.88	99.9
	$0.060 \rightarrow 0.0514$	33.1	1783.1	86.0	14.47	100.0

Table 12 shows how each hyperparameter changes the selected pruning depths. Increasing the temperature moves the mean first-pruning layer from 6.00 to 16.00, with MMStar ranging from 32.9 to 33.1, MME from 1774.8 to 1801.2, and POPE from 85.8 to 86.0. Stronger quadratic decay moves the mean from 13.45 to 12.31, while increasing the threshold scale moves it from 6.09 to 14.47. Thus, the hyperparameters predictably change when the first stage is executed; the default setting is

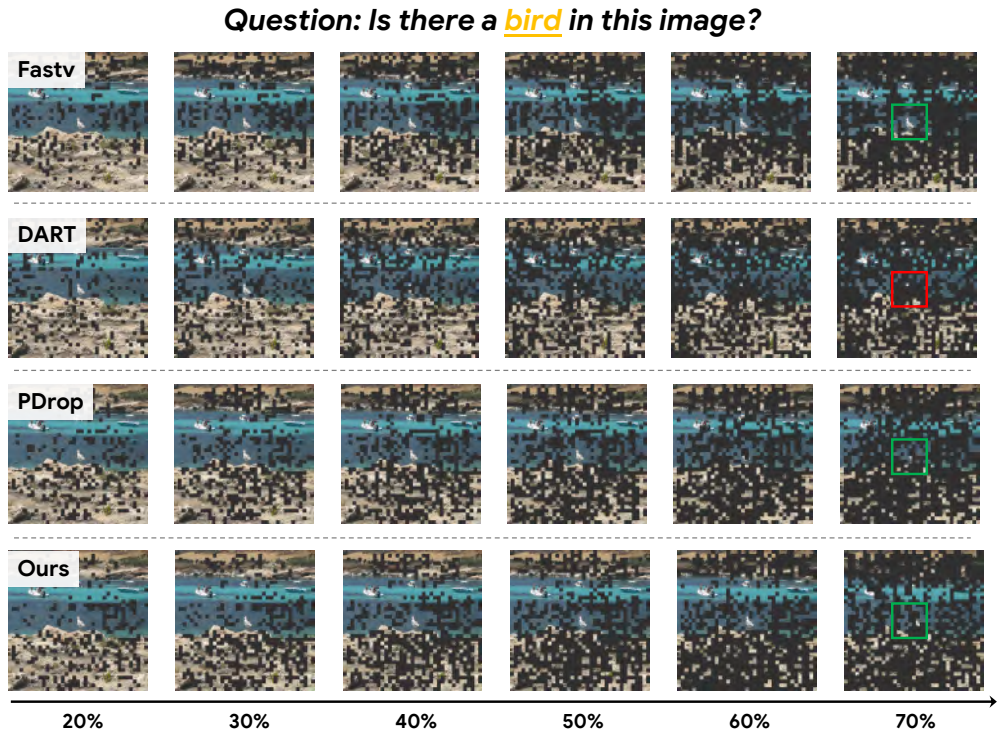


Figure 9: **Token-retention masks for a localized bird query.** Each row shows the visual regions retained by one reducer across the evaluated budgets.

the configuration used in the reported experiments rather than the best value for every metric in this sweep.

B.8 Additional Qualitative Visualizations

Figure 9 shows a localized recognition example in which the question refers to the bird. Across the evaluated budgets, HEAT's retained mask remains concentrated on the bird, while the comparison masks cover larger portions of the background. This is a representative retention pattern rather than evidence that all localized queries should be pruned at the same depth.

Figure 10 shows the full budget progression for the chart example used in the main paper. At the highest pruning ratio, HEAT retains both the vertical-axis values and the curve around 1975, while the comparison methods preserve incomplete or less query-aligned chart regions. The masks clarify why retaining both local numerical evidence and the broader curve trend is necessary for the correct answer.

Figure 11 shows an appliance query in a cluttered tabletop scene. Under the tighter budgets, HEAT continues to retain the appliance region, whereas the comparison masks assign more retained positions to other parts of the scene. The example illustrates the query-relevant regions preserved by the resulting depth and token-selection decisions without treating a single case as a general rule.

Figure 12 presents a query in which the phone lies beside a visually salient salad plate. HEAT keeps the retained mask centered on the phone and its surrounding context across reduction levels, whereas the comparison methods retain more of the nearby plate. This representative example shows the difference in retained regions under the same query and budget.

AI use statement

In this work, we used generative AI tools solely for language polishing of the manuscript. We have not used generative AI tools for any other tasks requiring disclosure. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

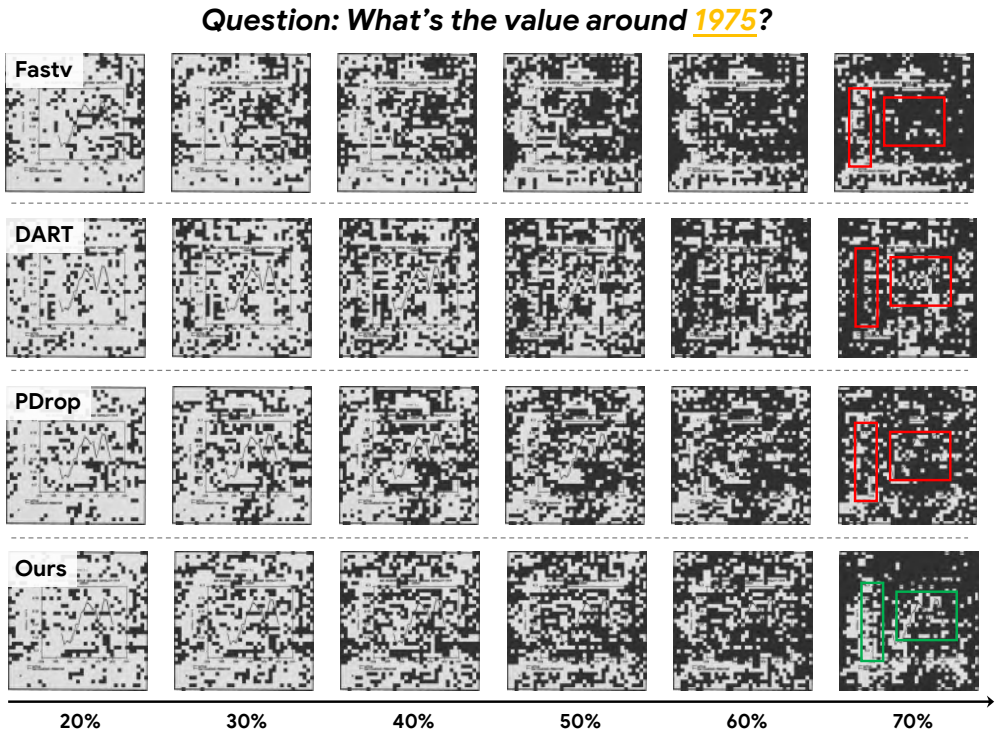


Figure 10: **Token-retention masks for a chart-value query.** HEAT preserves the vertical scale and curve trend around 1975 as the pruning ratio increases.

Ethics Statement

This work does not involve human subjects, personally identifiable information, or sensitive data. All datasets used are publicly available and were processed in accordance with their original licenses and terms of use. The authors declare no conflicts of interest or sponsorship.

Reproducibility Statement

We provide the complete source code in an anonymous repository: `https://anonymous.4open.science/r/HEAT-C08E`. The repository contains the implementation of our method, data preprocessing scripts, training and evaluation code, and documentation for reproducing the experimental results reported in this paper.

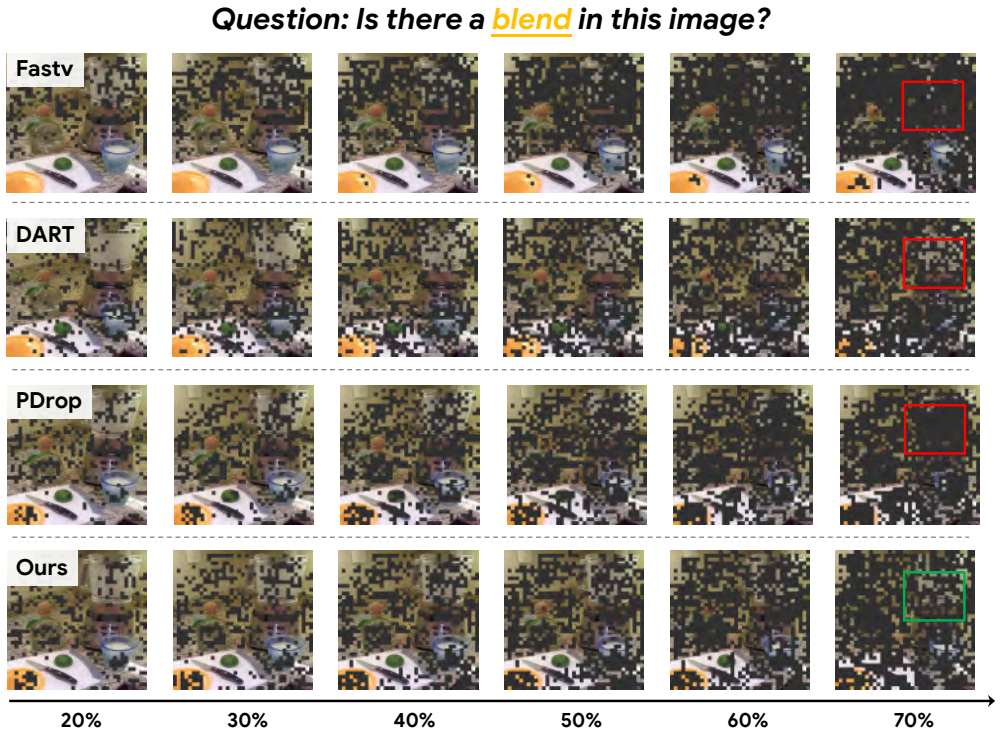


Figure 11: **Token-retention masks for an appliance query in a cluttered scene.** Masks compare the regions retained by each reducer across budgets.



Figure 12: **Token-retention masks for a phone query with a nearby salient object.** Masks compare the regions retained by each reducer across budgets.