HAC: Hierarchical Attention Consensus for Visual Token Pruning in Large Vision-Language Models

1st Jingru Li*
School of Computer Sciences
China University of Geosciences
Wuhan, China
jingruli810@cug.edu.cn

2nd Haowen Zheng*
School of Economics
Central University of Finance and Economics
Beijing, China
2023312303@email.cufe.edu.cn

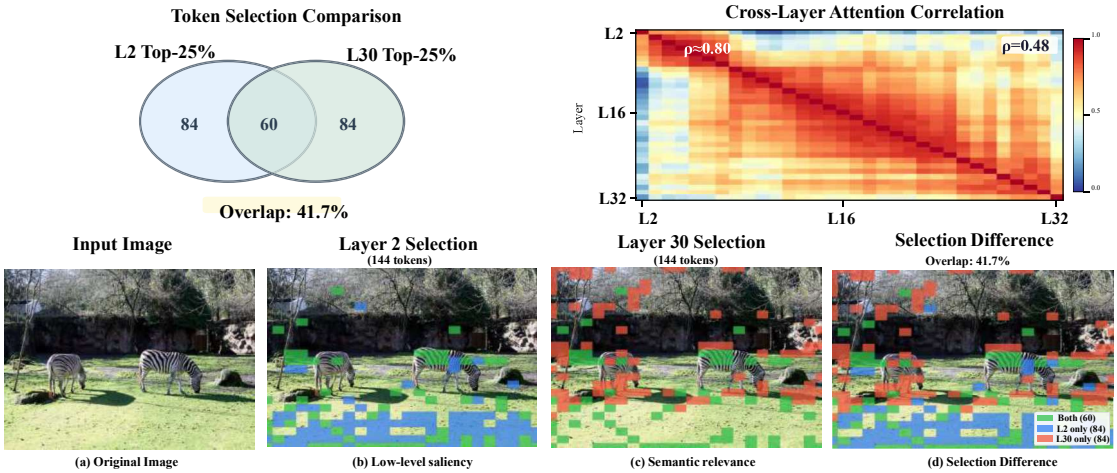

Fig. 1. Cross-layer attention divergence on a representative image: Layer 2 and Layer 30 select only 41.7% overlapping tokens ($\rho = 0.48$). **Green**: tokens selected by both; **Blue**: L2-only tokens; **Red**: L30-only tokens. The mean overlap across multiple images is 38.0% (Fig. 4).

***Abstract*—Visual token pruning has emerged as an effective strategy for reducing the computational cost of large vision-language model (LVLM) inference by decreasing the number of tokens processed in deep layers. Existing methods typically estimate token importance based on attention patterns from a single layer of the language model backbone, often selecting shallow layers for computational efficiency. However, we empirically observe that token importance rankings diverge substantially across network depth: the Spearman correlation between Layer 2 and Layer 30 attention distributions is only $\rho = 0.48$, with a mean token selection overlap of just 38% across diverse images. This cross-layer disagreement suggests that single-layer pruning may discard tokens consistently attended to by deeper processing stages. We propose Hierarchical Attention Consensus (HAC), a simple framework that aggregates normalized attention scores from multiple layers: shallow, intermediate and deep to identify tokens that maintain importance across the processing hierarchy. By preserving tokens with consistent cross-layer attention, HAC better captures tokens relevant to downstream tasks while filtering those with only layer-specific salience. The method requires no additional training and introduces minimal computational overhead. Experiments on vision-language benchmarks show that HAC consistently improves performance over single-layer baselines under the same computational budget, with gains concentrated on tasks requiring fine-grained visual understanding.**

***Index Terms*—vision-language models, token pruning, hierarchical attention, efficient inference, transformer representations**

*These authors contributed equally to this work.

I. Introduction

The emergence of large vision-language models (LVLMs) has enabled remarkable capabilities in visual understanding and reasoning [3]–[5]. These models typically encode images through a visual encoder and project the resulting representations into the token space of a large language model, enabling unified processing of visual and textual information [6], [7]. However, this architectural design introduces a substantial computational burden: a single image may be represented by hundreds of visual tokens, and the self-attention mechanism scales quadratically with sequence length [8], [9]. For instance, LLaVA-1.5 transforms a 336×336 image into 576 tokens [6], while higher-resolution variants may require over 2,000 tokens per image [7]. As LVLMs are deployed in increasingly demanding applications—video understanding, multi-image reasoning, and real-time interaction, the computational cost of processing visual tokens has become a critical bottleneck.

This challenge has motivated a growing body of work on visual token reduction, including pruning methods that discard less important tokens [12]–[14], merging methods that combine similar tokens [15], [16], and hybrid approaches [17], [18]. Among these, attention-based pruning has emerged as particularly attractive due to its simplicity: by monitoring which tokens receive attention during early processing stages,

these methods can make pruning decisions without additional learned parameters.

A common design pattern underlies attention-based pruning methods. These approaches observe attention distributions in shallow layers of the language model backbone, compute importance scores based on how much attention each visual token receives, and prune tokens with low scores before they propagate to deeper layers. FastV [12], a representative method, examines attention at layer 2 and achieves 45% FLOPs reduction with minimal performance degradation. Subsequent refinements have addressed various failure modes of shallow-layer scoring, including attention sink artifacts [13] and multi-cue selection [14]. Despite these advances, most methods still commit to a token importance estimate derived from a single processing stage.

In this work, we examine whether single-stage importance estimation is sufficient. Prior work in vision transformer analysis has established that representations evolve substantially across network depth, with different layers encoding qualitatively different information [20]–[22]. This suggests a natural concern: if attention distributions at layer l^* reflect that layer's particular representation, pruning decisions made at l^* may not align with the model's ultimate information preferences at deeper stages.

To directly quantify this concern, we measure cross-layer attention agreement across 500 diverse images (Figure 1). The results reveal substantial divergence: the Spearman rank correlation between Layer 2 and Layer 30 attention scores is only $\rho = 0.48$, and the two layers select only 38% overlapping tokens on average when each retains the top 25%. The correlation matrix (Fig. 3) exhibits a clear block-diagonal structure, confirming that different depth ranges constitute distinct processing regimes. Tokens selected exclusively by Layer 30 are systematically closer to the image center and show markedly higher overlap with semantic foreground masks compared to Layer 2-exclusive tokens (Table II). These findings motivate a pruning criterion that is not tied to any single processing stage.

Building on these observations, we propose Hierarchical Attention Consensus (HAC), a framework for visual token pruning that accounts for cross-layer attention divergence. The key insight is that token importance should be assessed through cross-layer consistency rather than single-layer magnitude. A token that consistently attracts attention across multiple hierarchical levels is more likely to carry information relevant throughout the abstraction process, while a token attended to primarily at one specific layer may reflect that layer's particular processing regime rather than global importance.

HAC operationalizes this insight by tracking the attention trajectory of each token across network depth. We sample attention distributions from layers spanning early, middle, and late processing stages, normalize each distribution for cross-layer comparability, and aggregate them to obtain a consensus importance score. Tokens ranking highly only at a single layer are down-weighted, while tokens maintaining consistent attention across the hierarchy are preserved.

HAC introduces no additional learnable parameters and requires only attention readout from a small number of strategically selected layers. The computational overhead is negligible, preserving the plug-and-play nature that makes attention-based pruning attractive. HAC can be integrated with existing frameworks as a drop-in replacement for single-layer importance estimation.

Our contributions are as follows:

- We empirically analyze cross-layer attention patterns in LVLMs and find that Layer 2 and Layer 30 exhibit only $\rho = 0.48$ Spearman correlation with 38% average token selection overlap, and that Layer-30-exclusive tokens show substantially higher semantic foreground coverage (58.2% vs. 31.4% IoU). These findings demonstrate that single-layer scoring is unstable across the processing hierarchy.
- We propose Hierarchical Attention Consensus (HAC), a training-free framework that aggregates normalized attention scores across multiple processing stages to identify tokens with consistent cross-layer importance.
- We conduct extensive experiments demonstrating that HAC consistently improves upon single-layer baselines across multiple benchmarks under equal computational budgets, with gains concentrated on tasks where preserving semantically salient tokens matters most.

II. Related Work

A. Visual Token Reduction

Modern LVLMs integrate visual perception with language reasoning by encoding images through a vision encoder (often CLIP [23] or SigLIP [24]) and processing the resulting tokens through a large language model backbone [4], [6], [7], [25]. The quadratic complexity of self-attention creates significant computational bottlenecks when processing hundreds of visual tokens per image.

Compression in Vision Encoder. Token merging methods [15] aggregate similar tokens based on feature similarity, while token pruning methods [27] discard less important tokens based on learned or heuristic scores. VisionZip [28] and LLaVA-PruMerge [29] combine both strategies.

Compression in Projector. Query-based compression (Q-Former in BLIP-2 [25]) uses learnable queries to distill visual information through cross-attention. Transformation-based methods employ pooling [30], [31], pixel shuffle [32], or convolution operations.

Compression in Language Model. FastV [12] pioneered attention-based pruning within the LLM backbone. FasterVLM [13] identifies that attention sink patterns, where attention concentrates on trailing tokens rather than semantically relevant ones, can corrupt importance estimates derived from the last text token; it addresses this through CLS-based scoring. HAC addresses a complementary but distinct failure mode: even with a well-chosen query position, importance scores from a single layer may not agree with those from other depths due to the evolving nature of transformer

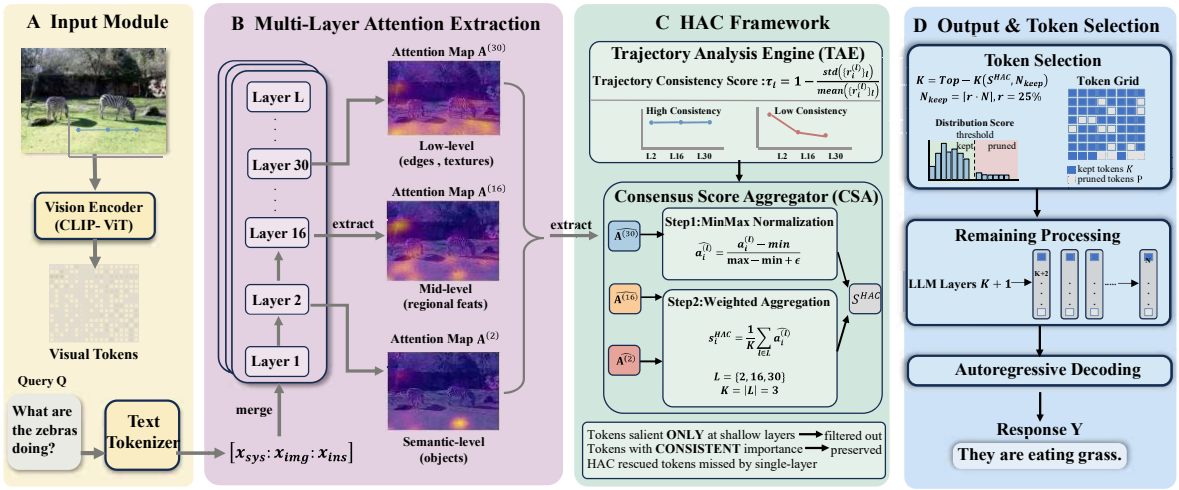

Fig. 2. **Overview of HAC.** HAC samples attention maps from multiple transformer layers spanning early (L2), mid-level (L16), and late (L30) processing. Token importance is normalized per layer and aggregated via cross-layer mean pooling. Tokens prominent only at a single layer are down-weighted; tokens with consistent cross-layer attention are preserved. Token selection occurs after layer L30; subsequent layers process the pruned token set.

representations. Subsequent works have further explored progressive multi-stage pruning [33], text-guided selection [34], adaptive compression [35], diversity-based criteria [18], and clustering-based reduction [17]. HAC is orthogonal to query-position fixes and can be combined with them.

B. Hierarchical Representations in Vision Transformers

A substantial body of work has analyzed the hierarchical structure of transformer representations. Raghu et al. [20] demonstrated that shallow ViT layers capture local patterns while deeper layers progressively form global, semantic representations [21], [22]. This depth-dependent evolution of representations implies that attention distributions at any single layer reflect the features encoded at that depth, motivating importance estimation that spans multiple layers rather than committing to one.

III. Method

As illustrated in Fig. 2, we propose Hierarchical Attention Consensus (HAC), a *training-free and plug-and-play* framework for visual token pruning in LLM-based multimodal models. Given an input image and text query, HAC extracts visual tokens using a frozen CLIP-ViT encoder and computes token importance by aggregating attention maps sampled from multiple transformer layers. Only a small set of tokens with consistent attention across layers is preserved and forwarded to subsequent LLM layers.

A. Preliminary: Attention-Based Token Pruning

Consider a vision-language model processing an image $\mathbf{I}$ through a visual encoder, producing N visual tokens $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$. These tokens are concatenated with text tokens $\mathbf{T}$ and fed into a language model with L transformer layers. At layer l, the self-attention computes weights $\mathbf{A}^{(l)} \in \mathbb{R}^{H \times S \times S}$, where H is the number of heads and S is the sequence length.

Attention-based pruning estimates token importance by aggregating attention received from query positions. Let $\mathbf{a}_i^{(l)}$ denote the aggregated attention for visual token i at layer l:

$$\mathbf{a}_i^{(l)} = \frac{1}{H} \sum_{h=1}^{H} \sum_{j \in \mathcal{Q}} A_{h,j,i}^{(l)} \tag{1}$$

where $\mathcal{Q}$ denotes query positions (e.g., the last text token). Existing methods such as FastV [12] select a single layer l^* (typically $l^* = 2$), rank tokens by $\mathbf{a}_i^{(l^*)}$, and discard the bottom $(1 - r) \times N$ tokens where r is the retention ratio.

B. Attention Trajectories Across Network Depth

The single-layer approach assumes that attention at layer l^* reliably indicates token importance throughout the full processing hierarchy. We formalize the cross-layer evolution of token importance through the *attention trajectory*.

[Attention Trajectory] The attention trajectory of token i is the sequence of its normalized attention scores across sampled layers $\{l_1, \ldots, l_K\}$:

$$\boldsymbol{\tau}_i = \left(\tilde{a}_i^{(l_1)}, \ldots, \tilde{a}_i^{(l_K)}\right), \quad \tilde{a}_i^{(l)} = \frac{a_i^{(l)} - \min_j a_j^{(l)}}{\max_j a_j^{(l)} - \min_j a_j^{(l)} + \epsilon} \tag{2}$$

where min-max normalization ensures cross-layer comparability.

A token whose normalized attention score remains consistently high across sampled layers is more likely to carry information relevant throughout the abstraction process, while a token prominent at only one depth reflects that layer's particular processing regime. HAC operationalizes this intuition by using the mean of normalized attention scores across layers (Eq. 3) as an efficient cross-layer importance estimate.

C. Hierarchical Attention Consensus

HAC selects tokens based on aggregated importance across multiple hierarchical levels. We sample attention from K layers spanning early processing ($l_1 = 2$), mid-level processing

Algorithm 1 Hierarchical Attention Consensus

Require: Visual tokens $\mathbf{V}$, retention ratio r, layers $\{l_1, \ldots, l_K\}$
Ensure: Pruned token set $\mathbf{V}'$
1: **for** $k = 1$ to K **do**
2: Forward pass to layer l_k, collect attention $\mathbf{a}^{(l_k)}$
3: Normalize: $\tilde{\mathbf{a}}^{(l_k)} \leftarrow \text{MinMaxNorm}(\mathbf{a}^{(l_k)})$
4: **end for**
5: Compute consensus: $\mathbf{s}^{\text{HAC}} \leftarrow \frac{1}{K}\sum_{k=1}^{K} \tilde{\mathbf{a}}^{(l_k)}$
6: Select top $\lceil r \cdot N \rceil$ tokens by $\mathbf{s}^{\text{HAC}}$
7: **return** Selected tokens $\mathbf{V}'$

($l_2 = 16$), and late processing ($l_3 = 30$). The HAC importance score for token i is:

$$s_i^{\text{HAC}} = \sum_{k=1}^{K} w_k \cdot \tilde{a}_i^{(l_k)} \tag{3}$$

where w_k are layer weights with $\sum_k w_k = 1$. We use uniform weights $w_k = 1/K$ by default. Tokens consistently ranking high across layers receive high consensus scores, while tokens prominent only at specific layers are down-weighted.

HAC is implemented as a drop-in modification to existing frameworks (Algorithm 1). Token selection occurs after the forward pass reaches layer l_K, so the first l_K layers process the full token sequence. Layers $l_K{+}1$ through L then operate on the pruned set of $\lceil r \cdot N \rceil$ tokens, reducing computation in the late-prefill stage and, more significantly, reducing the KV cache footprint throughout autoregressive decoding. The additional overhead introduced by HAC relative to FastV is minimal: storing attention from $K - 1$ extra layers and performing $O(KN)$ normalization and aggregation operations.

IV. Experiments

We conduct comprehensive experiments to validate HAC's effectiveness in improving token selection quality and downstream task performance. Our evaluation addresses four key questions: (RQ1) Does HAC improve performance over single-layer pruning methods? (RQ2) How significant is cross-layer attention divergence? (RQ3) What characterizes tokens preserved by HAC's multi-layer approach? (RQ4) How do different design choices affect the results?

A. Experimental Setup

We evaluate HAC on LLaVA-1.5-7B and LLaVA-1.5-13B [7], using CLIP-ViT-L/14 as the visual encoder (576 tokens per image) and Vicuna-7B/13B with 32 transformer layers. We compare against the full model, random pruning, FastV [12], and FasterVLM [13]. Evaluation spans captioning (NoCaps, Flickr30k), VQA (TextVQA, GQA, SEED), reasoning (MMMU, MMBench, SciQA), hallucination (POPE), and fine-grained evaluation (MME, MM-Vet). For each benchmark, we randomly sample up to 5,000 examples; datasets with fewer samples are evaluated in full. HAC uses layers $\{2, 16, 30\}$ with equal weights. All methods use pruning ratio $R = 50\%$ unless otherwise specified. Experiments are conducted on NVIDIA H100 GPUs.

B. Main Results

Table I presents performance comparisons across vision-language benchmarks. HAC consistently outperforms single-layer pruning baselines while maintaining the same 55% FLOPs budget.

LLaVA-1.5-7B Results. HAC achieves competitive performance with the full model across most benchmarks while reducing computational cost by 45%. Compared to FastV, HAC shows improvements on TextVQA (+4.67), GQA (+4.82), SEED (+3.24), and POPE (+3.91). The gains are more pronounced on hallucination detection (POPE) and fine-grained evaluation (MME), where HAC achieves 86.69% and 1498.7 respectively, approaching full model performance. Against FasterVLM, HAC demonstrates improvements on TextVQA (+3.32), GQA (+2.88), and MM-Vet (+0.86).

LLaVA-1.5-13B Results. Similar trends are observed on the larger model. HAC achieves 107.84 CIDEr on NoCaps and 65.74% on SEED, both approaching full model performance. Compared to FastV, HAC improves on TextVQA (+3.36), GQA (+4.14), SEED (+3.15), and POPE (+4.68). Against FasterVLM, HAC maintains advantages across most benchmarks, including TextVQA (+0.71) and GQA (+1.65).

The consistent improvements across diverse tasks—from captioning and VQA to reasoning and hallucination detection—confirm that cross-layer aggregation produces a more reliable token importance signal than single-layer scoring. Gains are larger for tasks where preserving semantically relevant tokens matters most (VQA, POPE), and smaller for tasks where coarse visual coverage is sufficient.

C. Cross-Layer Attention Analysis

Figure 3 visualizes attention correlation across all layers, computed over 500 COCO validation images. Adjacent layers maintain high similarity (e.g., L2 vs. L4, $\rho = 0.84$), but correlation decreases with distance, reaching a minimum of $\rho = 0.32$ between Layer 2 and Layer 12. A block-wise structure emerges (e.g., L12–L20), suggesting distinct processing stages. The final output layer (L30) shows only moderate correlation with the shallow input layer (L2, $\rho = 0.48$), indicating that single-layer scoring at L2 produces rankings substantially different from those that would arise at L30.

We further examine "rescued tokens"—those selected by Layer 30 but not Layer 2. Table II shows these tokens exhibit substantially higher overlap with ground-truth segmentation masks (58.2% vs. 31.4% IoU) and are located closer to the image center (124.7 vs. 186.3 px). Their lower edge response (0.38 vs. 0.74) suggests that Layer-2-exclusive tokens are more concentrated in high-edge-response regions. These empirical differences explain HAC's performance gains: preserving Layer-30-selected tokens retains visual content that single-layer methods discard.

D. Ablation Studies

Table III examines HAC's design choices. For layer selection, using only Layer 2 or Layer 30 yields suboptimal results (61.31% and 62.65% average). Combining both layers

TABLE I

PERFORMANCE COMPARISON ON VISION-LANGUAGE BENCHMARKS WITH $R = 50\%$ PRUNING RATIO. BEST PRUNING RESULTS IN **BOLD**, SECOND BEST UNDERLINED.

		Captioning		VQA			Reasoning			Halluc.	Fine-grained	
Method	**FLOPs**	NoCaps	Flickr	TextVQA	GQA	SEED	MMMU	MMB	SciQA	POPE	MME	MM-Vet
LLaVA-1.5-7B (Vicuna-7B)												
Full Model	100%	103.2	50.18	59.87	63.24	65.11	35.56	0.73	63.42	87.53	1510.7	29.78
1-13 Random	55%	82.6	38.91	42.16	48.72	51.34	28.47	0.58	52.89	74.82	1142.5	19.45
FastV	55%	100.8	47.94	53.47	57.00	60.24	**35.00**	0.71	62.07	82.78	1465.3	26.41
FasterVLM	55%	101.5	48.51	54.82	58.94	61.78	34.22	0.72	61.34	84.15	1482.1	27.68
HAC (Ours)	55%	**102.4**	**49.33**	**58.14**	**61.82**	**63.48**	34.33	**0.72**	61.68	**86.69**	**1498.7**	**28.54**
LLaVA-1.5-13B (Vicuna-13B)												
Full Model	100%	108.53	73.08	61.57	64.82	66.41	36.46	0.742	64.79	88.24	1531.37	31.26
1-13 Random	55%	87.34	56.27	45.31	51.96	53.83	29.18	0.618	54.23	76.57	1198.64	21.38
FastV	55%	106.28	73.46	56.83	60.24	62.59	36.74	0.726	63.17	83.18	1502.43	28.97
FasterVLM	55%	107.16	73.94	58.48	62.73	64.29	36.95	0.737	63.86	85.63	1518.76	30.14
HAC (Ours)	55%	**107.84**	**74.27**	**60.19**	**64.38**	**65.74**	**37.16**	**0.748**	**64.57**	**87.86**	**1526.39**	**30.87**

Note: NoCaps uses CIDEr score; MME uses total score; all others are accuracy (%).

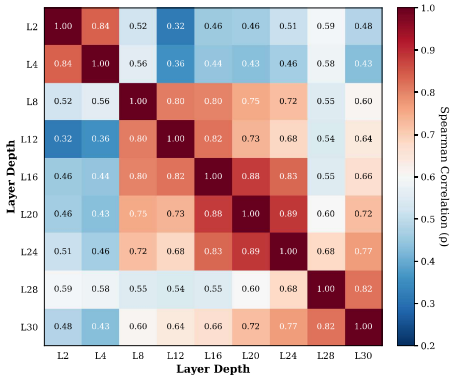

Fig. 3. **Cross-Layer Attention Correlation Matrix.** Spearman rank correlation of token attention scores between layers on 500 COCO validation images. Correlation decreases from shallow to middle layers (min $\rho = 0.32$), with L2 and L30 showing moderate correlation ($\rho = 0.48$), motivating multi-layer aggregation.

TABLE II

CHARACTERISTICS OF RESCUED TOKENS (L30 ONLY) VS. L2-ONLY TOKENS ON COCO DATASET. SEMANTIC OVERLAP: IOU WITH PANOPTIC SEGMENTATION FOREGROUND MASKS. EDGE RESPONSE: NORMALIZED SOBEL MAGNITUDE.

Metric	**L2-only**	**Rescued**	**Both**
Count per image	72	65	137
Distance to center (px)	186.3	124.7	98.5
Semantic overlap (%)	31.4	58.2	71.6
Edge response	0.74	0.38	0.52

improves performance (63.87%), and adding intermediate Layer 16 provides further gains (64.53%). Including additional layers shows diminishing returns (64.37%). For aggregation strategies, simple averaging performs well, while weighting deeper layers provides marginal improvements (64.65%). The minimum operation underperforms by being overly restrictive (61.84%).

Table IV evaluates HAC across different pruning ratios. Both HAC and FastV maintain reasonable performance at moderate pruning ($R = 0.25$), but HAC demonstrates greater robustness at aggressive rates. At $R = 0.6$, FastV shows degradation while HAC maintains stable accuracy. The improvement is particularly evident on GQA (+2.3%) and TextVQA

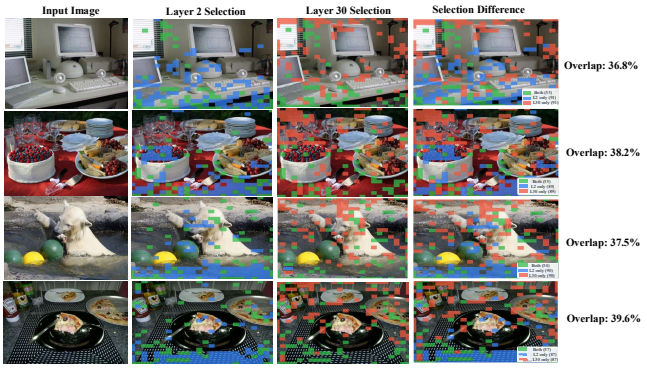

Fig. 4. Attention pattern visualization comparing Layer 2 and Layer 30 across multiple images. Columns: (a) Input image; (b) Layer 2 selection; (c) Layer 30 selection; (d) Selection difference. Token overlap ranges from 36.8% to 39.6% (mean: 38.0%).

TABLE III

ABLATION ON LAYER SELECTION AND AGGREGATION STRATEGY. DEFAULT CONFIGURATION IS HIGHLIGHTED.

Configuration	**TextVQA**	**MMMU**	**POPE**	**MME**	**Avg**
Layer Selection					
L2 only	53.47	33.11	82.78	75.86	61.31
L30 only	55.23	33.89	84.52	76.94	62.65
L2 + L30	57.38	34.05	85.91	78.12	63.87
L2 + L16 + L30	**58.14**	**34.33**	**86.69**	**78.94**	**64.53**
5 layers	57.96	34.28	86.54	78.71	64.37
Aggregation Strategy					
Mean (default)	**58.14**	**34.33**	**86.69**	**78.94**	**64.53**
Weighted (deep)	58.27	34.44	86.81	79.08	64.65
Minimum	54.91	32.76	83.45	76.22	61.84
Geometric mean	57.89	34.21	86.42	78.73	64.31

TABLE IV
PERFORMANCE COMPARISON ACROSS DIFFERENT PRUNING RATIOS ON LLAVA-1.5-7B USING A SMALLER SUBSET. HAC SHOWS INCREASING ADVANTAGE OVER FASTV AS PRUNING BECOMES MORE AGGRESSIVE.

Benchmark	R=0.25		R=0.3		R=0.5		R=0.6	
	FastV	HAC	FastV	HAC	FastV	HAC	FastV	HAC
POPE	85.2	85.1	84.9	**85.1**	84.8	**85.2**	84.1	**85.2**
GQA	60.8	**61.0**	60.5	**61.0**	59.8	**61.0**	58.8	**61.1**
TextVQA	58.0	**58.2**	57.9	**58.1**	57.0	**58.1**	56.1	**58.1**
Avg Δ	+0.10		+0.30		+0.87		**+1.79**	

TABLE V
INFERENCE EFFICIENCY COMPARISON ON LLAVA-1.5-7B. LATENCY MEASURED IN MILLISECONDS PER IMAGE, AVERAGED OVER 500 SAMPLES.

Method	Latency (ms)	Memory (GB)	Speedup
Full Model	2978 ± 351	13.95	1.00×
FastV	2910 ± 224	13.95	1.023×
HAC (Ours)	2924 ± 272	13.95	1.018×

(+2.0%), indicating that HAC's hierarchical consensus better preserves important tokens under aggressive compression.

E. Efficiency and Generalization

HAC introduces minimal computational overhead compared to FastV. Collecting attention from two additional layers and performing cross-layer aggregation adds approximately 14ms per image (0.47% of total inference time). The observed speedup (1.018× vs. FastV's 1.023×) reflects that token reduction in our implementation primarily reduces the KV cache footprint during autoregressive decoding and computation in late-prefill layers (L31 onward), while the full token sequence passes through layers 1–30 to enable consensus scoring. Table V reports detailed latency and memory statistics; memory footprint is identical across methods because pruning occurs mid-inference.

V. CONCLUSION

We presented Hierarchical Attention Consensus (HAC), a training-free framework for visual token pruning that addresses cross-layer instability in single-stage importance scoring for large vision-language models. Our empirical analysis finds that Layer 2 and Layer 30 share only 38.0% token selection overlap, and that Layer-30-exclusive tokens show substantially higher semantic foreground coverage than Layer-2-exclusive tokens (58.2% vs. 31.4% IoU). By aggregating normalized attention scores across multiple processing stages, HAC identifies tokens with consistent cross-layer importance, effectively rescuing tokens that single-layer methods discard. Experiments across diverse benchmarks demonstrate consistent improvements over baselines under equal computational budgets, with gains concentrated on tasks requiring fine-grained visual understanding.

REFERENCES

[1] X. Yue et al., "MMMU: A massive multi-discipline multimodal understanding benchmark," in *Proc. CVPR*, 2024.
[2] Y. Liu et al., "MMBench: Is your multimodal model an all-around player?," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024.
[3] OpenAI, "GPT-4V(ision) system card," OpenAI, Tech. Rep., Sep. 2023.
[4] J. Bai et al., "Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond," *arXiv preprint arXiv:2308.12966*, 2023.
[5] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "MiniGPT-4: Enhancing vision-language understanding with advanced large language models," in *Proc. ICLR*, 2024.
[6] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Proc. NeurIPS*, 2023.
[7] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," in *Proc. CVPR*, 2024.
[8] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. ICLR*, 2021.
[9] A. Vaswani et al., "Attention is all you need," in *Proc. NeurIPS*, 2017.
[10] M. Maaz et al., "Video-ChatGPT: Towards detailed video understanding via large vision and language models," in *Proc. ACL*, 2024.
[11] B. Li et al., "LLaVA-OneVision: Easy visual task transfer," *arXiv preprint arXiv:2408.03326*, 2024.
[12] L. Chen et al., "An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models," in *Proc. ECCV*, 2024.
[13] Q. Zhang et al., "[CLS] attention is all you need for training-free visual token pruning: Make VLM inference faster," *arXiv preprint arXiv:2412.01818*, 2024.
[14] B. Luan et al., "Multi-cue adaptive visual token pruning for large vision-language models," *arXiv preprint arXiv:2503.08019*, 2025.
[15] D. Bolya, C.-Y. Fu, X. Dai, P. Zhang, C. Feichtenhofer, and J. Hoffman, "Token merging: Your ViT but faster," in *Proc. ICLR*, 2023.
[16] D. Marin et al., "Token pooling in vision transformers for image classification," in *Proc. WACV*, 2023.
[17] M. Dhouib et al., "PACT: Pruning and clustering-based token reduction for faster visual language models," in *Proc. CVPR*, 2025.
[18] S. R. Alvar, G. Singh, M. Akbari, and Y. Zhang, "DivPrune: Diversity-based visual token pruning for large multimodal models," in *Proc. CVPR*, 2025.
[19] K. Tao, C. Qin, H. You, Y. Sui, and H. Wang, "DyCoKe: Dynamic compression of tokens for fast video large language models," in *Proc. CVPR*, 2025.
[20] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?," in *Proc. NeurIPS*, 2021.
[21] M. Naseer, K. Ranasinghe, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Intriguing properties of vision transformers," in *Proc. NeurIPS*, 2021.
[22] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel, "Deep ViT features as dense visual descriptors," in *Proc. ECCV Workshops*, 2022.
[23] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. ICML*, 2021.
[24] X. Zhai et al., "Sigmoid loss for language image pre-training," in *Proc. ICCV*, 2023.
[25] J. Li et al., "BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *Proc. ICML*, 2023.
[26] W. Dai et al., "InstructBLIP: Towards general-purpose vision-language models with instruction tuning," in *Proc. NeurIPS*, 2023.
[27] Y. Rao et al., "DynamicViT: Efficient vision transformers with dynamic token sparsification," in *Proc. NeurIPS*, 2021.
[28] S. Yang et al., "VisionZip: Longer is better but not necessary in vision language models," in *Proc. CVPR*, 2025.
[29] Y. Shang et al., "LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models," in*Proc. ICCV*, 2025.
[30] X. Chu et al., "MobileVLM V2: Faster and stronger baseline for vision language model," *arXiv preprint arXiv:2402.03766*, 2024.
[31] L. Yao et al., "DeCo: Decoupling token compression from semantic abstraction in multimodal large language models," *arXiv preprint arXiv:2405.20985*, 2024.
[32] Z. Chen et al., "How far are we to GPT-4V? Closing the gap to commercial multimodal models with open-source suites," *Science China Information Sciences*, 2024.
[33] L. Xing et al., "PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction," in*Proc. CVPR*, 2025.
[34] Y. Zhang et al., "SparseVLM: Visual token sparsification for efficient vision-language model inference," *arXiv preprint arXiv:2410.04417*, 2024.
[35] J. Zhang et al., "p-MoD: Building mixture-of-depths MLLMs via progressive ratio decay," in*Proc. ICCV*, 2025.